MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY
LEXINGTON 73, MASSACHUSETTS

VOlunteer 2-3370

August 15, 1958

Mr. R. W. Bemer
Programming Systems Department
International Business Machines Corp.
112 East Post Road
White Plains, New York

Dear Mr. Bemer,

Your interest in our work on the Lincoln Keyboard is appreciated and we will try to keep you informed. Since the "Lincoln Writer" is still in development, there is no published information available, but I am sending you a copy of a memo I received from Jack Gilmore subsequent to showing him your letter.

The "Lincoln Writer" will be used for direct in-out computer applications and also as a tape preparation facility. A prototype is under construction. Jack Gilmore's memo should give you the information you require.

The code values for the "lower case" keyboard were chosen so that numerals correspond to codes and the letters are in alphabetical order. The key positions are standard. The "upper case" keyboard can indeed be grouped in any way you like. The code values, however, are determined by the key slugs on the typewriter. In most cases the wishes of the engraver were respected. For example, it was easier to put the larger symbols over the numerals since there is more room on the slug. In most applications, the code values are not critical since they will be translated by the program anyway.

I hope your questions have been answered satisfactorily. If you have any further questions do not hesitate to let me know.

Yours truly,

Alexander Vanderburgh, Jr.

Alexander Vanderburgh, Jr.

AV/md

NMAH 310/1

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY
LEXINGTON 73, MASSACHUSETTS

August 14, 1958.

VOlunteer 2-3370

To: Alex Vanderburgh

From: Jack Gilmore

Subject: The Lincoln Writer

Before giving specific details, I believe Mr. Bemer will be interested in the events leading up to our decision to design our own symbols and program preparation unit.

To begin, one first must relate our experience with the TX-0 computer. Although it was a simple machine it possessed a magnetic core memory of 65,536 registers. When we started to debug our first programs (which by the way seemed lost in that forest of cores) the usual post mortem routines were read into the rear of memory. It did not take long to realize that by utilizing a direct input-output console typewriter we could communicate with these post mortem routines and request information regarding our programs while we were still at the computer. What followed was a simple but useful utility system of routines which allowed us to examine and modify our programs in programming language;-search for particular constants or instruction words; introduce intermediate stopping points in our programs which switched control to the utility system thereby allowing us to examine what had happened up to a certain point and then continue on; etc. All this was done because of a large memory and a direct input output typewriter.

The type of research programs being operated on TX-0 required that the computer be used more as an experimental tool rather than a calculator for equations whose solution methods were known. The ability to stop, examine and modify data as well as debug programs became extremely useful and soon we wondered how we ever got along without this technique of communication.

With the coming of the TX-2 computer and its efficient but complicated logic we knew that our present "ability to express" would limit us in communication with the computer. While TX-0 was still in possession of the big memory we wrote a program which allowed us to simulate a typewriter with 200 characters. This was done by using the display scope and a photodiode light pen. The program was called the TX-0 Scopewriter.

The Scopewriter developed into more than a simulated typewriter but as a typewriter, we discovered that although the desired number of symbols was much larger than what an ordinary typewriter could provide, the ability to place these symbols into superscript and subscript position as well as the normal position increased the "expressability" considerably.

From the information gained from the Scopewriter plus endless discussions, we decided that what we wanted was a typewriter tape preparation unit which could be used to prepare programs away from the computer and which contained all the necessary symbols and typewriter functions needed to express arithmetic, mathematics, symbolic logic, algorithms, and plain English. In addition we wanted a typewriter attached to the computer with the same ability. Since such a unit did not exist we decided to undertake the job ourselves.

The unit we have decided upon contains the following:

1. An electric typewriter with 88 characters capable of producing all 88 characters in super, normal and subscript positions. It is capable of decoding electrical input signals for all characters and typewriter functions.
2. An auxiliary keyboard with a key button position for every corresponding character and typewriter function of the typewriter. Each keybutton is capable of producing a coded electrical signal.
3. A mechanical paper tape reader.
4. A paper tape punch.
5. A central control unit for configuring the components of the unit according to the desires of the typist.
6. Connections for attaching it to a computer.

The name of the unit is, "The Lincoln Writer".

As a tape preparation unit it will be used to:

1. Type and punch a program or data tape.
2. Read a paper tape and obtain a printed copy and a second punched tape (if desired).
3. Duplicate typewritten punched paper tapes or binary tapes.

As a direct input-output computer console unit it will:

1. Allow a programmer to type requests and send them into the computer and if desired, obtain a punched paper tape copy of the requests.
2. Allow the computer to type replies and questions on the typewriter and if desired, obtain a punched paper tape copy of these responses.
3. Although highly unlikely the mechanical reader can be used to type and transmit requests to the computer via a previously prepared tape.

In other words the unit can be used in the same way as when it is off the computer with the added feature that the computer can monitor what is going on and communicate with the unit accordingly.

The task of selecting only 88 characters was a difficult one and took a great deal of time on the part of all members of the Logic Section of the Computer Research and Development Group (63). The final selection is a compromise on all our parts but we feel that a great many of the absent symbols can be produced by combining some of the "88" via the backspace.

The overbar and underbar will not cause the platen to be shifted to the right one space position. This allows the typist to avoid backspacing when using these two symbols to mark another symbol. The same is true for the box and the circle symbols. With the exception of the English alphabet and a few large symbols all other characters can be circled. The box can be used on all characters. These four symbols allow us to modify characters and thereby change their meaning.

The Auxiliary keyboard was included in the unit for the convenience of the typist. A normal keyboard would have required switching from upper case to lower and from superscript to normal to subscript positions. The auxiliary keyboard has a keybutton for each character and provides the case change automatically.

When a keybutton on the auxiliary keyboard is selected the keyboard's circuitry logic will sense what the case of the typewriter is and if it is not in the case of the selected character then the necessary case code will be generated first and then the selected character's code will be generated.

For engineering convenience, the coding was assigned so that all codes above 57 octal would not require a case change. These include the typewriter functions and a few special codes which will not produce a printed character.

The spare codes have been assigned keybuttons on the auxiliary keyboard and will be used as special signals to the computer when the unit is attached to it.

I am enclosing a copy of our auxiliary keyboard, the typewriter's keyboard and list of the characters. The control panel is not completely decided upon but my proposal is included alongside the keyboard panel plate. A simple schematic of the Lincoln writer is sketched in the upper left corner of the copy. Also it may be of interest to point out that the keybuttons will have their character's code engraved in octal notation on the front side of the button facing the typist. We hope this will assist the typist in remembering codes when examining the tape she is preparing. One last item. We have color coded the keybuttons to assist the hunt and peck artists like myself. The green keys are English alphabet characters; yellow, arithmetical; brown, symbolic logic; orange, Greek; red, punctuational; grey and black, functional, and special.

The keyboard of the typewriter will never be used except for maintenance purposes. Therefore the typewriter will be mounted in a well so that the auxiliary keyboard will fold across the typewriter keyboard and be as close to the typed copy as possible so that the typist will see the copy easily as she types on the auxiliary keyboard.

The typewriter key slugs and arm levers for the 88 characters are being manufactured for us by the IBM Electric Typewriter Division, Lexington, Kentucky. The designer assigned to the job is Howard Kettler. He was quite helpful in the final design of the symbols.

The typewriter which is to be used is an IBM typewriter modified by Soroban Engineering Inc., Melbourne, Fla. Their product is called a Computerwriter and has an excellent coding and/or decoding package. We have also contracted them to produce the auxiliary keyboard and the mechanical reader.

I have probably given too many details but I've met Bemer a few times, one of which was at a demonstration of TX-0, and I suspect he might be interested in them.

Jack Gilmore

Jack Gilmore

JG/kr

NMAH 310/1

C-502
SSD - 37
Page 1 of 3

SHARE XI, San Francisco
September 11, 1958

TO THE MEMBERS OF SHARE:

The following 64-character alphabet is suggested, for consideration by the proposed SHARE ad hoc Committee on Algebraic Languages, as a possible point of departure for hardware implementation of the Reference Language being developed by the participants in the ACM-GAMM Conference on an International Algebraic Language.

The Reference Language was intended to be a link between "publication" and "hardware" languages. The purposes of this proposal are to show that an essentially explicit form of the Reference Language itself can be written in a six-bit-per-character alphabet, and to urge that SHARE develop, and encourage hardware realization of, such an alphabet for computer input and output.

The suggested alphabet omits lower-case letters and does not use overprinting. It includes all ten decimal digits with unique "zero" and "one"; all twenty-six letters of the English alphabet, with unique "I", "𝒪", and "X"; twenty-seven special characters including the I.A.L. floating decimal designator; and "blank" (which is the same as "space").

The fact that 64 characters can be represented explicitly by a six-bit code is important in connection with adaptation of present-day computer data media. Among these are six-active-track tape (paper as well as magnetic), twelve-row punched cards, twelve-active-track magnetic tape, and shaped-beam CR tube printers which typically use a stencil having 64 apertures in an 8 x 8 array.

With the exception of the lower-case English letters, which are replaced by a modulo-two-shifting scheme (see note (e) below), the entire Reference Language[*] can be written directly in this alphabet, with no transliteration, becoming thus a Hardware Language.

Notes on some of the more obviously debatable choices:

(a) Numeral "one" uses the exaggerated headstroke of the German script "1". Letter "I" uses exaggerated top and bottom bars.

(b) Letter "𝒪" is larger than "o" (zero) and includes a flourish at the top as when written as an English script capital letter. These distinctions are, clearly, a matter of personal taste. Better ones may well be proposed; in any event, I feel that some obvious distinction should be made, and that the presently common use of "Ø" to represent the letter "O" is confusing to the uninitiate and is harder to write than "𝒪".

---

[*] J. Backus to SHARE members, "Zurich Conference on a Universal Programming Language," C-471 (SSD-35), August 14, 1958, pages 30-35.

(c) Letter "X" has slightly exaggerated horizontal marks at each extremity to emphasize its "capital letter" nature. It is larger than the bold-face multiplication cross "×".

(d) The Boolean connectives "and" (cap ⊓), "or" (cup ⊔), and "not" (inverted check mark ¬) use right-angle corners for ease of writing and reading manuscript.

(e) The intended use of the single quote (which was chosen, instead of some other vertical-axis-symmetric character, for compatibility with the SHARE 709 alphabet) is as a single-level bracket to simplify scanning of delimiter words.

(f) The tentative Reference Language includes six equality-inequality operators = ≠ > ≥ < ≤ . It seems that either ">" and "≥" or "<" and "≤" might be omitted without undue inconvenience; "≠" may be replaced by "¬ =". The two or three characters thus made available might be used in several ways:

1. The two-character representations of the assignment delimiter " : = " and the procedure delimiter " = : " might be replaced by the special single characters ":=" and "=:" . I would consider this option less desirable than the following one.

2. Inclusion of the asterisk "*" and dollar sign "$" would complete the SHARE Standard 704 and 709 alphabets as used by SAP, SCAT, and FORTRAN.

3. A "single-character-shift" designator which can be stacked would permit not only an indefinite extension of the alphabet for infrequently-used characters, but also an easy way to provide several dozen unique two-character abbreviated control words for frequent use. If the character "¤" is used thus to signify "shift next character", then, "¤B" might be used in place of " 'BEGIN' " or "BEGIN", etc.

(g) The rightward arrow "→" substitution delimiter is considered by many Americans to be less appropriate than a leftward arrow "←". The former is included here because, at Zurich, the Americans yielded on this item to the Europeans.

Your serious consideration of this matter is solicited. I would hope that SHARE can standardize, at an early date, on a 64-character, 6-bit alphabet.

Yours truly,

Herb Bright

H. S. Bright (WB)

Attachment: TABLE I, "A Proposed 64-Character Alphabet for Computer Input-Output"

P.S. "$" and "¤" will be of additional usefulness to those interested in business data processing.

		BASIC	OPTIONAL	
			Per Zurich	Modified
NON-DELIMITERS	digits	012 through 9		
	letters	AB through HIJ through NOP through XYZ		
	blank	(no mark; same as "space")		
DELIMITERS	operators	+ - × / = > ≥ ⊓ (and) ⊔ (or) ≡ (equivalence) ¬ (not)	≠ < ≤	A $ (instead of "≠", use "¬=")
	separators	. , : ; ⏨ (basis of ten)	→ substitution	← substitution
	brackets	( ) parentheses) [ ] subscript)(can be ↑ ↓ exponentiation)nested) ' single quote (cannot be nested)		⌘ shift next character (can be stacked)

TABLE I.

"A Proposed 64-Character Alphabet for Computer Input-Output"

(HSB 9/11/58)

A SYSTEMS APPROACH TO INTEGRATION OF AUTOMATIC DATA PROCESSING AND COMMUNICATIONS

Captain W. F. Luebbert
U. S. Army Signal Research & Development Laboratory
Fort Monmouth, N. J.

Introduction

It is a safe prediction that automatic data processing will have an impact upon communications in the next ten years equally as revolutionary as its impact on business machines in the past ten. This impact will be felt across the board in the amounts, kind, and even in the fundamental concepts of the nature of communications.

Traditional concepts of communications have dealt with the transfer of information from one location to another, preferably with little or no change of form. However, the integration of automatic data processing equipment and techniques into communications systems provides new dimensions for communications planning. Data processing equipment can calculate, sort, merge, compare, collate, correlate, abstract, filter, change format, and otherwise manipulate information. Thus, the integrated communications-data processing systems of the future will be capable of not only transferring information from one location to another, but also of transforming it in content and format.

Many symptoms of this impending revolution are apparent. Operational communications systems, which should be the last to feel the impact of such a revolution, are already feeling preliminary rumbles of it. For example, in the past three years requirements for data transmission have grown from nothing to about 1/5 the total communications load for the Army Command and Administrative Network. New installations such as communications and message switching centers are experimenting with the utilization of computers as control elements to handle complex problems of routing, cryptographic system assignment, switching control, and directory functions.

New equipment developments are emphasizing the integrated family approach. For example, the Army's FIELDATA program makes no distinction between communications and data processing equipment. Both utilize the same input, output, and storage devices; the same voltages, impedance levels, codes and other common interconnection characteristics; and the equipments are designed to perform functions so overlapped that it is often difficult to determine whether a particular item should be called an item of data processing or an item of communications equipment.

Most important of all, painstaking, thorough, and detailed analyses of new system concepts are being made to create systems, such as the Army's proposed UNICOM, which will take advantage of the new capabilities and techniques becoming available.

Impact of Data Processing Upon Amount of Communications Required

Experience to date shows that the introduction of data processing equipment is often accompanied by an explosive increase in communications requirements. In part this is due to the voracious appetite of data processors for current, up-to-date information in applications which take advantage of their abilities to process vast amounts of information economically. In part it is because their tremendous speed of processing compared to the speed of human processors dramatically changes time factors in an overall data handling cycle. For example, if a certain report requires forty-five days to prepare using non-automated techniques, a delay of three or four days in transmission of the raw data to prepare it because the data is sent by mail is hardly noticed. However, if a data processor requires only one day to prepare the report from the raw data, this same three or four day delay in the mail becomes the overwhelming factor determining overall report preparation time, and an obvious bottleneck. Obviously, if the report is an important one, badly needed for effective planning, there is a strong tendency to demand better communications service to eliminate the bottleneck. Hence, considerable amounts of data once transferred by mail are frequently added to the electrical communications requirements.

This mushrooming of communications requirements is not, of course, an inevitable result of the adoption of data processing. On the contrary, one of the most attractive features of data processors to the communicator is their ability to process and reduce raw data to a more concentrated form, boiling out those portions which are irrelevant, redundant, or insignificant so that the communications system and its human users need not be burdened. If a data processor is used to reduce data to 1/10th its raw form, it may then be used to reduce communications required to transmit that data by the same factor. Thus data processors acting as data reduction devices located at an entry point of data into a communications system may be a particularly valuable tool for preventing large sources of raw data from overloading a communications system with excessive demands for service.

Of course, the same increased speed of processing by machines which disrupts the balance of many processing-communications cycles and thus increases the demand for electrical communications may under other conditions have the opposite effect. If the key factor in a situation is the

1959 IRE National Convention Record

STANF_RD LIBRARY
JUL 6 1959
ENGINEERING

PART 4

ENGINEERING
JUL 9 1959
LIBRARY

Sessions Sponsored by
IRE Professional Groups on

Automatic Control
Electronic Computers
Information Theory

at

the IRE National Convention, New York, N. Y.

March 23-26, 1959

The Institute of Radio Engineers

achievement of a fixed delivery deadline and the processing time is reduced, then the processing time saved becomes available for communications. For example, if a fifteen day deadline exists and fourteen of those days are required for manual processing of data, then only one day is available for transmittal of the information and electrical transmission may be necessary to meet the deadline. However, if an automatic data processor can prepare the information in two days, the remaining thirteen days allow ample time for transmittal of the information in more routine fashion by mail or messenger, thus reducing the demand for electrical communications.

A thorough and properly conducted analysis of the complete information flow must be undertaken before selection of a processing scheme. The effects of the processing scheme on the communications requirements of the overall system may be a key determinant of the method of data handling and the type and amount of equipment used.

Which of these factors become dominant in a particular case when data processing equipment is introduced is not so much a function of the equipment used as it is of the concepts and techniques of employment used. The equipment will have an impact upon communications requirements, but it is only a tool, and the user and his method of employment of this new and powerful tool will, in the final analysis, determine what its impact will be.

Impact of Data Processing on Communications System Concepts

Automatic data processing is a tool - a powerful tool for handling, manipulating and processing data. If one is to treat automatic data processing as a tool, one should be prepared to describe its fundamental capabilities. Since communications is itself a tool, it is instructive to attempt to describe its fundamental nature in a similar frame of reference and compare the two. The results of such a simple analysis lead to a viewpoint which has important implications on system and equipment design.

The fundamental capabilities of data processing equipment can be described as the ability to transform information into more desirable or useful forms. Using the same viewpoint, the fundamental capability of conventional communications equipment can be described as the ability to transfer information to a more desirable or useful location. Of course, from an abstract/logical viewpoint a transfer is merely one special kind of transformation, one in which only physical location is changed. Combining these descriptions, one is led to the concept of a generalized data handling system capable of performing generalized transformations, of which conventional one-location data processing would be one special case and conventional no-processing data transmission would be another special case. In such a system there would be no fundamental distinctions between data processing and data transmission, but only distinctions of convenience based upon application, use and design emphasis.

Is such a concept a reasonable one, and does it have any practical utility? The answer to the first part of this question will be examined in detail in this paper. The answer to the second part of the question - whether this concept has practical utility will be determined by the success in actual use of the equipment and system concepts it generates, the concepts explained in this paper which are receiving initial implementation in the FIELDATA family of equipments.

An examination of the information flow and manipulation in typical data processing and data transmission equipments shows almost immediately that there is a very considerable mixture of functions going on in both types of equipment. A very considerable amount of the activity going on inside any computer or data processor consists of simple transfers of information from one part of the processor to another. Whereas, within data processing equipment a major part of the activity is concerned with the generation, manipulation and other processing of information used for control, supervisory and error reduction purposes. In many cases nearly identical operations go on in both data processing and data transmission equipments, with the differences, if any, being matters of design emphasis based upon application and use.

Many practical cases of this similarity are immediately obvious, particularly in the area of devices used for data entry and output. For example, computers frequently use paper tape readers similar to those used in teletypewriter transmitter distributors, and paper tape punches that could easily be used in teletypewriter reperforators. Similarly the idea that kinds of input-output devices such as card readers and punches and magnetic tape transports which are widely used for data processing can effectively be adapted for use of communications lines is also being exploited in equipments such as the IBM transceiver, the Collins Kinetape, etc.

FIELDATA Equipment Concepts

FIELDATA emphasizes this kind of exploitation to the extreme, particularly encouraging it by assuring that interconnection of semi-autonomous equipment modules be made in accord with common standards without distinction whether these equipments are conceived primarily for computer-associated or transmission-associated functions. This makes it possible for the same data terminals to operate not only with data processing-type inputs and outputs such as computers, paper tape, magnetic tape and IBM cards; but also for it to operate with real time weapons system data and with telegraphic data.

Control circuitry is so devised that pure binary as well as alphanumeric (alpha beticnumeric) data may be handled. Since any digital code, be it Baudot (teletypewriter) code, Holerith (IBM card) code, or any of a wide variety of computer codes may be represented in binary bit-by-bit form, the FIELDATA devices have the potential of transmitting or handling any type of digital data. Thus, they could be used with digitized voice, digitized facsimile, or other types of digitized analog signals.

The use of common standards, codes, and standard data rates makes possible the kind of data transmission equipment concept shown in Figure 1. This concept leads naturally to a division of the sub-assemblies of data transmission equipment into three kinds:

1) Input-Output Transducers: These are devices for converting information from some human or machine usable form such as paper tape, magnetic tape, punched cards, analog electrical voltages, strokes on a keyboard, etc. into digital form.

2) Transmission Transducers: These are devices for converting data in digital form into appropriate signals for transmission over radio, wire or other kinds of propagation media.

3) EMBOLIC* Equipment: This equipment, normally inserted between input-output transducers and transmission transducers, is used primarily to perform control and supervisory functions, error detection and/or correction, buffering, and/or speed conversion, code conversion, or encryption necessary for proper system operation of the data transmission equipment. The functions of embolic equipment are information processing functions. Inputs and outputs will both be digital in form although supplementary analog information may also be available particularly in some kinds of error control schemes. A general purpose computer is potentially a very powerful and flexible type of embolic equipment, but the necessary functions can often be performed much more economically by specialized equipment.

This division of sub-assemblies may be a physical division into separate items of equipment, into semi-autonomous parts of a single equipment (either in separate boxes or in a single box) or it may be merely conceptual with no physical implementation.

Obviously maximum flexibility and adaptability can be achieved if any input-output transducer can operate with any transmission transducer. This permits one to tailor an equipment to meet the requirements of a particular situation. It permits one to choose the in-out device suitable for the user's most convenient data form (paper tape, magnetic tape, IBM cards, etc.) independently of the nature of the transmission facility available. One can then choose the transmission transducer on the basis of the transmission medium used (v.h.f. radio relay, h.f. radio, loaded cable, carrier transmission on wire, etc.) independently of the user's data form. Then join the two together taking advantage of common standards of interconnection or intercommunication between modules to create a well tailored combination.

The question then arises, "Why embolic equipment?" Certainly by proper choice of common interconnection or intercommunications characteristics one can minimize requirements for code conversion, buffering, speed conversion, etc. Ideally one should be able to join the input-output transducer to the transmission transducer without embolic equipment, so why have it? The answer is that there are three important functions in a data terminal which seem convenient to separate from both in-out transducers and transmission transducers:

1. Communications Supervision
2. Error Control
3. Cryptographic Security

Supervisory functions are conveniently separated from the input-output transducers and the transmission transducers because supervisory requirements may be strongly influenced by both. Error control requirements may also be strongly influenced by both the in-out data and the transmission situation, and may even in some cases not be desired at all. Thus, it is best treated as a module which can be tailored to the paired requirements of in-out and transmission or completely omitted. Cryptographic security is conveniently a separate module so that it may be omitted in those cases where security is not required or desired.

Primary and Secondary Data

In order to discuss supervisory activities conveniently it is desirable to make a distinction between two kinds of information which flow through a communications system:

1) Primary Information: That information which a user wishes transferred to another location, that is, the information to be communicated.

2) Secondary Information: That information added to the primary information either by the originator or by equipment of the communications system which is used to perform functions of supervision, routing, error control and related activities necessary or desirable to permit the primary data to be effectively communicated. This information is used by communications equipment and personnel and is normally of no use or interest to the ultimate recipient of the primary information.

*EMBOLIC - A coined term from the Greek embolisimos - to put between or insert. This word is also used in medical, astronomic and ecclesiastic literature to describe other specific kinds of intercalations.

A basic requirement for maximum flexibility and adaptability is that the user, who enters data at an input device and receives data at an output device, need not be required to exercise judgement or knowledge, to perform special activities, to use, or to interpret secondary information. This, in turn, makes it desirable to isolate secondary information from the input-output devices, which are the users' point of contact with the communications system. A convenient way of establishing and enforcing this isolation is the creation of a distinct embolic module which generates, receives, interprets, and/or acts upon secondary information and passes on action requirements derived from this secondary information over local control lines to its associated input-output transducer and transmission transducer. It is important to note that such an equipment need not have any ability to interpret or act upon primary data, and thus its operation can be made completely independent the coding used for primary data so long as there is a unique method of distinguishing primary from secondary data.

Fieldata Code

In the FIELDATA family of equipments the segregation of primary and secondary data is accomplished in a particularly simple way by providing separate transmission symbol "alphabets" for primary and secondary data. The basic unit of data in FIELDATA codes is a six-bit character. For primary data this may be merely a six bit block with no specific meaning assigned to specific bits within it, and no restrictions on the permissible code combinations which may be transmitted. If this information is alphanumeric (alphabetic-numeric) in form so that specific bit meanings must be assigned for electric typewriters, printers and similar input-output transducers to operate, then the FIELDATA alphanumeric code given in Appendix "A" is used. A similar but distinctly different six bit FIELDATA supervisory data code has also been created. It is given in Appendix "B". In those situations where both FIELDATA primary data and FIELDATA supervisory code data might appear on the same six lines, the same six channels of tape, etc. a seventh bit is added to provide unique distinction. This seventh bit is normally a binary "one" for primary data and a binary "zero" for supervisory data. If, on a particular recording medium such as paper tape or on a particular kind of transmission channel, adequate reasons exist for tagging the difference between these two symbol alphabets in some other way, then other methods could be used. All secondary information except error control redundancy is handled by the FIELDATA supervisory data code in FIELDATA equipments.

Error Control

In some situations transmission errors may create important reasons for using other tagging and error protective redundancy schemes. Errors which cause loss or change in supervisory information such as dialing information, message headings, start and end of message indications and so forth may completely disrupt the proper functioning of a communications system. Thus they may require protective redundancy many times more powerful than the simple parity used in the basic eight bit form. However, compared to this secondary data primary data may be capable of tolerating considerably higher error rates. For example, if the data is English text the inherent redundancy of the language may permit significant corruption without loss of intelligibility. Numerical data may consist of successive observations of a physical phenomenon in such a form that inconsistant data may be deleted and ignored, or it may be protected by numerical checks similar to those used by accounting systems to detect bookkeeping errors. In a situation where error rates were severe it might not be desirable to apply as powerful error control to this less demanding primary data as to secondary data; thus one might desire to use methods of differentiating between primary and secondary data which are more resistive to corruption by errors than the simple tags used in the basic eight bit form together with different error control schemes for primary and secondary data.

This kind of differentiation illustrates a fundamental assumption about error control which is part of the FIELDATA concepts: Specific error control requirements should be determined by the nature of the data and its use. As mentioned above there may be cases where primary data can tolerate frequent errors with little loss of usefulness; but there are other cases, such as the transmission of computer programs, where very low error rates are required. In some cases if errors are above the desired maximum rate occur and they are detected, the erronious information may be deleted without significant harm; in other cases they must be corrected. In some cases where correction is required it may be deferred and handled by a service message, in other cases it must be corrected before the data is released. And so forth.

Notice that all these requirements are determined by the use of the data, and that these demands remain the same regardless of the transmission path and types of transmission equipment through which the data may be required to flow. However, the occurrence of errors is anything but independent of transmission factors. Although errors do occur in input-output devices and embolic equipment, by far the most variable and difficult to control errors normally occur in the transmission portion of a data link. These errors are often quite variable in frequency and interrelated in occurrence. They are quite strongly dependent upon the nature of the propagation medium and the kinds of disturbances to which it is subject. For example, a h. f. radio link subject to severe fading and multipath disturbances could hardly be expected to have the same error problems as a wire and cable link subject to intraceable crosstalk and impulse-type switching noise. Furthermore, for a given propagation medium and kind of noise and disturbances, the frequency and interrelationships of errors are strongly dependent upon the characteristics of the transmission transducers used. For example, if amplitude modulation is used one would expect different errors than if frequency shift keying were used; if sequential transmission of short bauds on a single sub-carrier is used one would expect different error problems than if parallel transmission of long bauds on multiple sub-carriers is used; and if sampling or non-integrating types of detection are used one would expect different error problems than if full integrating detection schemes were used.

The number, variability, and difficulty of measurement and analysis of the various factors which contribute to the frequency and interrelationships of transmission errors or particular circuits is staggering. At the present time the state of the art is such that only crude estimates of frequency of error can be made for typical equipments when exposed to disturbances other than gaussian noise, and that practically nothing can be estimated in advance about the interrelationships of errors under practical conditions of impulse noise, cross talk, propagation variations, etc. This is particularly unfortunate because the effectiveness of the various digital error control schemes available is strongly dependent upon the interrelationships of errors. For example, a simple parity check is capable of detecting single errors but not double errors. If errors occur randomly, double errors will seldom occur and this very simple check will be quite powerful. Thus if the bit error rate is 10^{-4}, a simple parity check will reduce the undetected error rate to 10^{-8}. On the other hand, if errors tend to be clustered a parity check will be rather ineffective. Thus if the conditional probability of a second error immediately following the first is 0.5 and the bit error rate is 10^{-4}, then a parity check will reduce the undetected error rate to only 0.5×10^{-4}. Given knowledge of the interrelations checks can be designed which give high protection with a minimum amount of checking equipment. Unfortunately this knowledge is usually unavailable.

If the most important sources of errors **is** a data communications link, is it proper to incorporate the major error control features of the link in the transmission transducers? The FIELDATA concepts answer this question with a resounding NO! Why? The key reason is that while the occurrence of errors and the raw error rate characteristics are determined primarily by transmission factors, the error requirements are determined by the use of the data. The means and types of error control appropriate to a particular situation obviously depend upon the interaction of these factors. However, it is a fundamental modularity principle of FIELDATA that the characteristics of transmission transducers should as nearly as possible be independent of the details of user input-output characteristics and data employment.

If one were to incorporate the error control features into the transmission transducers and one desired to provide p classes of controlled error service to users with q modulator/demodulator assemblies, then one would require p times q types of complete transmission transducers. The obvious answer is to separate modularly, making the error control module an item of **embolic** equipment. This allows one advantage of similarities in requirements and raw error characteristics among the different situations to reduce the variety of equipments to be constructed.

In view of the present difficulties in predetermining the specific error characteristics of transmission transducers prior to construction and test, it also permits construction of new transmission transducers and their use with existing error control embolic equipments until the specific error characteristics of the transducer can be measured and new error control embolic equipment designed if necessary to meet user requirements with the measured transmission error characteristics.

In addition to these practical advantages of placing error control responsibilities in embolic equipment modules rather than in transmission transducer modules, there are conceptual advantages associated with maintaining the simplest possible information flow patterns and division of activities among the three basic kinds of assemblies.

In general transmission transducers pay no attention to the information content of the digital information they convert to modulated transmission form, neither knowing nor caring whether the data is primary data or secondary data, whether it is redundant or irredundant, or what code or codes it uses. In contrast to this embolic equipments normally act as information processing devices. Thus, in supervising a transmission link embolic supervisory equipment act on sensory information received from transmission transducers and in-out transducers and generate, process, and interpret secondary supervisory information to control the overall operation of the communications link. The error control problem is exactly parallel. Acting on information about user requirements (from the in-out transducer side) and information and sensory information about transmission errors from the transmission transducer side, error control equipment is required to generate, process, and interpret error control information using it to control (often via supervisory operations) the overall operation of the communications link in such a way as to control its errors. Thus it is obvious that from the information processing viewpoint the performance of error control as an embolic function has a close parallel to other embolic functions and is distinctly different from the functions otherwise performed by transmission transducers.

Specific FIELDATA Equipments

Among the FIELDATA equipments being developed in implementation of these concepts are the following:

Transmission Transducers

AN/TSQ-32	600-1200 bit/sec	Stelma Corp.
AN/TSQ-33	600-2400 bit/sec	Collins Radio
AN/TSQ-35	19,200 b bit/sec	Bendix-Pacific

General Purpose Computers

MOBIDIC	Sylvania
BASICPAC/LOGICPAC	Philco
INFORMER/DATA COORDINATOR	I. B. M.

In-out Transducers

Electric Typewriter	Smith-Corona
Paper Tape Reader	Smith-Corona
Paper Tape Punch	Smith-Corona
High-Speed Printer	Anderson-Nichols
Paper Tape Transport	Ampex Corp.
Magnetic Tape Transport	Ampex Corp.
Tacden	Aeronutronics Systems

Special Embolic Equipment

CV-689	Cryptosecurity Adaptor	Collins Radio
CV-690	Control Equipment	Collins Radio
CV-691	Data Concentrator	Collins Radio

The transmission transducers of the FIELDATA family are limited in number. However, the choice of the 75 x 2^n pattern of data rates permits widespread augmentation by minor modification of existing or developmental teletypewriter multiplexed transmission equipments. In addition future expansion is simplified by the fact that future equipments will be able to utilize the same embolic and input-output equipments, and will thus be cheaper to develop than data transmission equipments which require development of embolic and/or input-output equipment as part of the same package. Expected future expansion will place greater emphasis on transducers for radio circuits.

The AN/TSQ-32 is a transmission transducer capable of accepting data from a standard FIELDATA connection and of transmitting it serially as frequency shift modulation of a single sub-carrier over a standard voice channel. Transmission rates are 1200 bits/sec (1500 words per minute) over good quality circuits, and 600 bits/sec over poorer circuits.

The AN/TSQ-33 is a transmission transducer capable of accepting data from a standard FIELDATA connection and of transmitting it as 8 channels as synchronous phase quadrature modulation of four sub-carriers over a standard voice channel. Transmission rates are 2400 bits/sec (3000 words per minute) over good quality circuits, with 1200 and 600 bits/sec available for use over poorer circuits. This equipment is essentially a militarized, miniaturized, FIELDATA compatibized version of the Kineplex TE-206.

The AN/TSQ-35 is a transmission transducer capable of accepting data from a standard FIELDATA connection and of transmitting it by amplitude modulation of 8 subcarriers orthogonally spaced in frequency and time. The transmission baseband is the 48 kc band between 12 and 60 kc used for military and commercial cable and carrier circuits. Representative circuits are the Army spiral-four and associated AN/TCC-7, 8, and 11 equipments; and commercial types N and K carrier equipments. In addition to point to point full duplex operation this equipment has special features for multiple station "common net" round-robin type operation.

In addition to these transmission transducers specifically designed for FIELDATA a wide variety of existing military and commercial teletypewriter transmission equipments can be used for FIELDATA transmission with only minor modification. Examples of such equipment are the AN/FGC-54 capable of transmission and diversity reception of FIELDATA information at 2400 bits/sec over long-haul 3 kc radio circuits using 32 channels each operating at 75 bits/sec. Another example is the AN/FGC-29 potentially capable of transmission and diversity reception FIELDATA information at 1200 bits/sec using 16 channels each operating at 75 bits/sec. Yet another example is the AN/TCC-30 potentially capable of transmitting 1200 bits/sec using 16 channels operating at 75 bits/sec.

The FIELDATA computers form the most complete and well balanced portion of the FIELDATA family.

The FIELDATA computers may act as either input-output transducers or as embolic equipments for data transmission purposes, having the ability to accept, process, and emit either primary or secondary data. All are designed for direct operation with data transmission equipment. In their employment the MOBIDIC, which was the first machine to have its characteristics frozen, is the least capable machine data transmission-wise, having serious restrictions on its use of the supervisory code functions. The most flexible in its employment of data transmission will be the DATA COORDINATOR, a newer equipment which will have the capability of terminating a large number of data transmission circuits simultaneously and which will have a number of special capabilities and console positions to facilitate its use as a facilities coordination processor for an integrated communications/data processing system.

All the FIELDATA computers are general purpose processors of modular design and great flexibility. Their relative computation speeds are shown on the bar chart. All are designed for field use with operation and maintenance simple enough for field personnel. The largest, MOBIDIC, mounts in a semi-trailer van, while the others mount in shelters which can be carried on a truck. It is interesting to note that the minimum assembly which forms a fully operational stored program computer of the BASICPAC/LOGICPAC type or the INFORMER/DATA COORDINATOR type weighs in either case less than 150 pounds subdividable into 50 pound or smaller packages. However, these central processors are normally used for data processing purposes in vehicular assemblies which far exceed these weights because of auxiliary equipments such as multiple tape transports, special real-time input-output units, additional magnetic core memory modules, data transmission equipment, input-output converters, etc. For example, a vehicular BASICPAC would augment the central processor with four magnetic tape transports, a high speed paper tape reader and punch, an AN/TSQ-33 transmission transducer, and other auxiliary equipment.

The input-output transducers of the FIELDATA family serve double duty as computer input-output devices and as data transmission input-output devices. The group under current development constitutes a minimum group of general usage items, a number of which are in only partially militarized form. This minimum group will be augmented by future field teletypewriter equipments which will utilize FIELDATA code, and by advanced equipments now under study to provide specialized input-output capabilities. Since the items are mostly quite conventional, detailed descriptions are omitted here in order to save space.

The specialized embolic equipments in FIELDATA provide cryptographic security, interconnection of input-output and transmission transducers, and tie-in of FIELDATA circuits to existing teletypewriter circuits.

Three major items of special embolic equipment are being developed for FIELDATA. The CV-689 is a special cryptosecurity adaptor which permits an existing type of security equipment to be inserted just before the transmission transducer in any FIELDATA data transmission assembly, thus providing cryptographic security.

The CV-690 is a device which provides supervisory control, error control, and synchronization buffer facilities for connecting paper tape or magnetic tape units to the AN/TSQ-32 or 33. Although future plans call for similar specially militarized embolic equipments for other kinds of input-output equipment such as card equipment, none is now under development. However, minor modifications of commercial [illegible] card equipment will permit non-[illegible] [illegible] of AN/TSQ-33 equipment for card transmission, and at least some versions of [illegible] will include card equipment which, through the computer, can reach transmission facilities.

Although FIELDATA concepts make provision for a wide variety of error control and supervisory control systems only the particular system used in the CV-690 will actually be used initially except for computer-to-computer transmission, since other embolic devices using different error control or supervisory control schemes are now under development. It is expected that when experiments do determine the actual frequency and interrelationship of errors for particular transmission transducers that more effective schemes will be devised. However, the very simple and easy to implement two dimensional (interlaced) parity error detection scheme followed by request-back or rerun type error correction used in the CV-690 is particularly easy to implement, and is expected to suffice for initial testing.

The CV-691 Data Concentrator is a device designed to bridge the gap between existing large scale 60 and 100 wpm teletypewriter facilities and FIELDATA equipment. Although all FIELDATA processors have normal provision for a paper tape reader which accepts teletypewriter tape as well as one for FIELDATA tape, there exists a significant need, especially at locations where FIELDATA processors might not be available, to accept multiple channels of teletypewriter information and convert it into FIELDATA form to take advantage of FIELDATA transmission transducers and error control, permit recording on FIELDATA magnetic or paper tape, permit printing on FIELDATA high speed printers, simplify entry into FIELDATA computers, etc. The CV-691 accepts up to 25 (or 50) teletypewriter inputs, stores the information in a buffer core memory (made up of the same memory planes as used in MOBIDIC), assembles it into message blocks, converts it into FIELDATA form, applies the same error and supervisory control as the CV-690, and emits the data in FIELDATA form at rates up to 2400 bits/sec (3000 words per minute). The receive side performs the inverse functions.

It is expected that as more and more of the voice and other analog communications systems convert to pulse code modulation and other digital forms that additional types of FIELDATA embolic equipment will be required to perform the error control, supervisory functions, and buffering/synchronization necessary to tie input-output transducers to their digital bit streams.

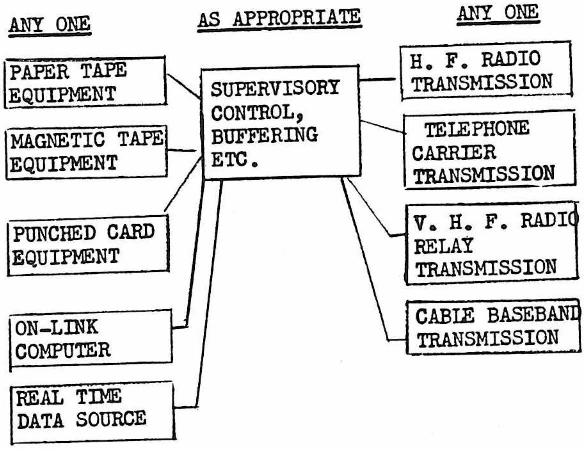
The FIELDATA family is an attempt to create an integrated family of data transmission equipments to meet Army needs. Though it lacks many of the features and equipments of an ideal family of data transmission equipments, will make available in experimental quantities by the end of 1960 the first integrated family of equipments for experimental establishment of a truly integrated communications/data processing system. The first system in which data processing and communications equipment both utilize the same input-output and storage devices, the same voltages, impedance levels, codes and other common interconnection characteristics; and in which the equipments are so designed that in many cases the only way to determine whether a device is used for communications or data processing is to look at its specific application in the system.

APPENDIX A

TABLE I
FIELDATA ALPHANUMERIC CODE (6 bit form)

$D_3D_2D_1D_0$	I_2I_1 00	I_2I_1 01	I_2I_1 10	I_2I_1 11
0 0 0 0	Master Space ≀≀	K	)	0
0 0 0 1	Upper Case ≀	L	-	1
0 0 1 0	Lower Case ▨	M	+	2
0 0 1 1	Tab. ◇	N	<	3
0 1 0 0	Car. Ret. ❮	O	=	4
0 1 0 1	Space △	P	>	5
0 1 1 0	A	Q	-	6
0 1 1 1	B	R	$	7
1 0 0 0	C	S	*	8
1 0 0 1	D	T	(	9
1 0 1 0	E	U	"	'
1 0 1 1	F	V	:	;
1 1 0 0	G	W	?	/
1 1 0 1	H	X	!	.
1 1 1 0	I	Y	,	Special ▭
1 1 1 1	J	Z	⊕ Stop	Backspace

FIGURE 1
DATA TRANSMISSION EQUIPMENT CONCEPT.



APPENDIX B

TABLE III
FIELDATA SUPERVISORY FUNCTION CODE (6 bit form)

$D_3D_2D_1D_0$	I_2I_1 00	I_2I_1 01	I_2I_1 10	I_2I_1 11
0 0 0 0	(Master Space)	(K)	Blank	RCA
0 0 0 1	(Upper Case)	(L)	Dial 1	RCB
0 0 1 0	(Lower Case)	(M)	Dial 2	RCC
0 0 1 1	(Tab.)	(N)	Dial 3	EOBK
0 1 0 0	(Car. Ret.)	(O)	Dial 4	EOB
0 1 0 1	(Space)	(P)	Dial 5	RCE
0 1 1 0	(A)	(Q)	Dial 6	RCF
0 1 1 1	(B)	(R)	Dial 7	RCG
1 0 0 0	(C)	(S)	Dial 8	RCH
1 0 0 1	(D)	(T)	Dial 9	RCJ
1 0 1 0	(E)	(U)	Dial 0	RCK
1 0 1 1	(F)	(V)	SOB	RCL
1 1 0 0	(G)	(W)	MDA	RCM
1 1 0 1	(H)	(X)	MDB	RCN
1 1 1 0	(I)	(Y)	MDC	RCP
1 1 1 1	(J)	(Z)	MDD	DELETE

NEW DATA DEVICE DEVELOPED

will provide multiple, long-distance access to computers

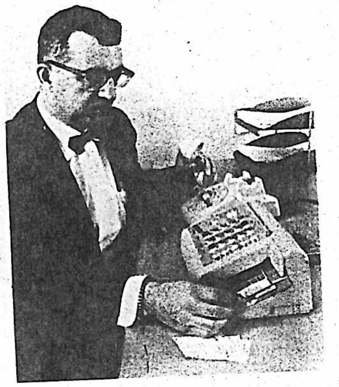
A further stride in mechanization of time-consuming clerical routines of business and government was announced recently by the Western Electric Company at a demonstration of a new invention designed for use with telephone company Dataphone service. Dataphone service provides for the transmission of data over regular telephone connections. Thus, this new device, which has not yet been named, will provide multiple access, unlimited by distance, to modern, high speed computers.

The development was sparked by a search on the part of the manufacturing and supply unit of the Bell System, the Western Electric Company, for better ways of ordering supplies from the company by employees of Bell telephone companies. Other Bell applications are being considered.

An experimental model is currently being demonstrated by engineers to several Federal agencies and to some 35 guage directly to the data processing center with no intermediate steps. Thus, wasteful manual duplication would be avoided. Moreover, Dataphone service permits information formerly mailed to be flashed over regular telephone lines with a consequent saving in time.

Sevebeck said the electronic apparatus at a receiving center could accept data sent over telephone connections from many scattered points, by means of small, low-cost transmitters like the model he demonstrated Information flowing from these points into the center could be recorded automatically on paper tape or punch cards, then transferred directly into standard data processing machines.

The multiple input feature of the system which is inherent in Dataphone service would help to solve the stubborn problem of making large and costly computers justifiable in the solution of many business problems. Such information



An engineer prepares to implement Bell System's new Dataphone development. After telephoning the receiving center, he is ready to send business data through the card reader. This can be done by using small punched cards or the numbered pushbuttons.

At receiving point, girl puts Dataphone equipment on the telephone line by throwing a key. This connects the distant sender to the card punch (background) which will record transmitted information on cards, for direct use in data processing equipment.

business machine manufacturers to acquaint them with its capabilities and to interest them in manufacturing and marketing the transmission device for general public use. Field tests are planned to validate proposed use of the device.

Laurin L. Sevebeck of Western Electric, a co-inventor of the equipment, explained that the new facility now being perfected could not be made available to potential users by business machine manufacturers for several months to come.

He described the problem which, it is hoped, use of the new device with Dataphone service will solve. The company, he said, receives about 500,000 large and small orders monthly from telephone companies for tens of thousands of items carried in stock. More direct methods of feeding orders to the distribution centers and into data processing machinery would improve efficiency and speed up all phases of filling orders, inventory control and accounting.

The new device, operating in conjunction with Dataphone service, is designed to move input of data away from the data processing center to hundreds of originating points. From there data would be transmitted in machine lan- is usually assembled, reprocessed and then checked manually for insertion into the computers or other devices. Upon considering computer speeds, the question has been how to supply enough information to them conveniently and economically in the case of business operations.

The first large-scale trial of this development will get underway this Spring with the installation of transmitting and receiving units in the states of New York and Illinois. Telephone company men using the new equipment to transmit orders to Western Electric will have small identification cards, containing code punch marks. They will also have a supply of similar cards with punched code numbers describing items they would have occasion to order.

The telephone company employee using the service would insert his identification card in the card reader – part of the transmitting device. This automatically establishes his authority to order materials and specifies delivery information. Next, he would insert cards corresponding to the items he wished to order. The quantity of each item would be transmitted by pressing numbered keys on the top of the device. ●

Circle 141 on Reader Service Card.

v5 #2 Mar/Apr 1959

MICRO
INIATURIZATIO
FOR
IGITAL
COMPUTERS

resistors, 4 capacitors, 4 diodes and
are contained in the complete flip-f

the Hughes Research & Development Lab-
tories, dramatic reduction in component
is symbolic of the advances in the digital
nputer art...advances which are preparing
digital computer for its role as a system
trol in space.

n addition to developing new digital cir-
s and components, Hughes engineers are
ving into other digital computer space mis-
tasks such as navigation guidance and
trol, communications processing and dis-
, vehicle control, mission programming.

n intensive Hughes program in digital
nputer systems has created openings for

Automatic Data Processing in the Tactical Field Army

A. B. CRAWFORD†

THE use of digital computer techniques has become mandatory in the whole realm of ballistic missile design and guidance. And we are rapidly approaching the era in which computer techniques similarly might well become a necessity in many phases of ground combat. The feasibility of applying these techniques in a tactical warfare environment has been established, and the Department of the Army has a high priority program underway to place a prototype Automatic Data Processing System (ADPS) into its tactical organization within the next four years. This paper will explain the design objectives of this system and the progress to date in the applications area, and describe the extension of several techniques required to make the system function.

The technical design objectives may be summarized in about three key phrases—integrated, common-user service center, employing general-purpose hardware. The system in employment has been likened to the widespread tactical communication system, that is, a network of switching centers operated by specialists but tying together all users or customers, largely through the use of common-user trunks and switchboards. More specific design guidelines include the following:

1) Common-user facilities with a minimum number of single-user processors;
2) Integrated data files;
3) Simple foolproof input devices;
4) Interrogation-immediate reply;
5) GP, building-block hardware;
6) Automatic operation;
7) Flexible output.

The tactical ADPS will consist of several elements. (See Fig. 1.) Simple, foolproof input devices will be located at the source of data input. Connecting the inputs to the data-processing center and centers with each other is the Army's area communication system. According to the work load and applications being processed, each center will employ one or more general-purpose computers. Finally, a whole range of output devices is available according to the type of output desired, or rather required, by the commander to facilitate his making a sound decision—volatile but rapid video-display graphical or map-overlay form, or hard-copy reports.

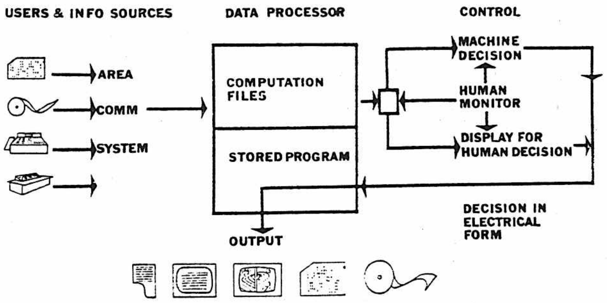


Fig. 1—The elements of the tactical ADP System are located in such a fashion as to minimize the burden on the users of System operation.

Functionally, the system is portrayed as an integration of subsystems according to the military staff function mechanized. The Army's traditional categorization is the well-known G-1, 2, 3, and 4 or Admin., Intelligence, Operations, and Logistics. In each of these categories, however, the uses of information are similar—that is, immediate reaction, longer range planning, and historical preparations. Our system design must be tailored to meet the rigorous specifications of high speed for the immediate reaction use as well as the capacity for large amounts of file maintenance and processing.

Looking at the over-all system configuration, one sees a far-flung complex of data processors and data transmission links, the magnitude of which is presently unique. As is well known, the tactical Army's command structure runs from the Company headquarters to Battle Group, Division, Corps, Army, etc. It appears desirable—and in fact a necessity—to install a data-processing capability within each headquarters from Battle Group back. The physical size and processing abilities will of course increase as we proceed toward the rear of the Combat zone. In fact, at Corps and Army more than one computer will be linked together to make up that data-processing center.

Now, everything described to this point might be called long-range planning. The title of the system discussed is ARMYDATA, and the general time-frame is beyond the immediate future. Nevertheless, we are guided in the immediate systems developments by these objectives. The remainder of this paper will describe the Army's project to attain a prototype ADP System as an intermediate step toward reaching the ultimate.

Slightly less than two years ago, a major project was launched within the Army to place into operation as soon as possible a system which could support the ever-increasing military requirements for mobility and effectiveness in combat. Because of the long lead-time inherent in the R & D cycle for complex digital computers, the decision was made to parallel the development of the system's hardware and the system's concepts and applications. (The full description of the so-called FIELDATA family of equipment has been left to the companion paper prepared by Captain Luebbert of the Signal R & D Laboratories.)

---

† Automatic Data Processing Dept., U. S. Army Electronic Proving Ground, Fort Huachuca, Ariz.

At the time the decision was made to proceed on the hardware, a plan of action was laid out whereby the entire Army's methods for information handling in the field were to be scrutinized in great detail. The ultimate objective was a detailed description of each user's data-processing requirement with the more immediate by-product of streamlining the manual methods.

Those persons associated with business data-processing systems are familiar with the peculiarity of the systems problems involved in mechanizing integrated business procedures of any magnitude, and here I use the word "system" to mean "procedural" structure. We propose to integrate some 50 or more separate procedures (for as many customers) from concepts developed initially in an isolated fashion. That is, the logistics applications are being derived through systems analysis by the logistics specialists, the combat intelligence applications by G-2 specialists, and so on. To take full advantage of the great power of automatic data processing, we must mold or integrate these individual requirements into a realistic yet imaginative working system which meets the needs of the diverse staff requiring its services. The types of applications to be satisfied again can be broken down into the traditional areas of Operations, Intelligence, Logistics, and Admin. A different categorization according to data-processing techniques might be Combat Control (random access), Combat Support (batched processing), and Combat Computation.

Approximately twenty separate study groups are analyzing some seventy separate potential ADP applications. To assure that these studies will be more directly usable, an instructional booklet has been distributed setting forth a very rigid report format and content. The fact is that the area specialists are not experienced in the techniques of systems analysis, so we designed the booklet more or less to lead the study group "by the hand" through the steps of an ADP application study or systems analysis. Further, through contracts and the use of our own systems analysts, we are providing technical assistance to the study groups, particularly as they reach the phase of proposing an automated procedure for their problem.

Upon completion the individual study reports are subject to a technical review to determine feasibility, completeness, and the amount of dependence on other applications. Analysts return to the agency which conducted the study to fill in details and to eliminate ambiguities. The object of the next step is to define the application to the detail required for programming and to derive certain workload data to be used in the over-all system design.

About this time, too, the preparation of digital models of the system and its parts is to begin. This, I believe, implies the underlying approach to our whole problem: simulation! We shall be utilizing all means and all levels of digital simulation techniques. First of all, interpretive simulation must be employed to permit the preparation and checking out of the computer programs prior to the delivery of the militarized computers. A successful simulation of the MOBIDIC on the IBM 709 has been completed, and so far we have put through successfully a few math subroutines, an intelligence filing and retrieval experiment, a combat surveillance target acquisition simulation, and a limited payroll problem. To extend this technique we are attempting now to develop a simulator generator which will enable us to experiment inexpensively when altering machine parameters or order codes. (This latter is an attempt to lay the groundwork for the later definition of the advanced system characteristics and computer designs.)

Before the detailed design of the prototype system is launched, a subset of the available application areas will be selected for implementation. Only then can we say specifically how comprehensive the FIELDATA system will be in covering tactical procedures. Since it is recognized that the hardware components are experimental in nature, in order to meet a tight schedule without a crash program the FIELDATA system will itself be experimental in nature.

Parallel to the detailed programming of each application will be the use of a digital model of the system information flow to permit a prediction of the needed data rates, alternate routing, and potential bottlenecks. A model is being used now using the 709 which simulates a general army area communications complex. Probability distributions and Monte Carlo techniques are employed throughout, from the preparation of a message entering the system to human switchboard operator actions and to message processing as it progresses toward its destination. A statistical analysis routine produces model-run data reduction and also recommends a sample size according to our level of confidence desired.

To explain the next type of simulation, first I shall outline the organization of the ADPS Test Facilities at the Proving Ground. Computer test facility is around a large-scale computer center, specifically the IBM 709. Herein all of the simulation work will be conducted, and herein will be the means for conducting controlled environmental tests for predicting the validity of our proposed computer procedures. The field-test facilities

---

are for the pur
urally be empl
and model res
we are to utili
the 709 will b
subjected to e
ment; a single
supplemented
system. (See F
the data sink
absorb outpu
erated.

The Comp
ruary, 1959.
are complete
several of the
on at least tw
type system
this coming
reported by
sion, and pro
our objective

In summar
and futuristic
to provide t

FIELD
ing an
velope
family is th
tional disti
processing.
and technic
merger emp
communica
a high deg
other data
degree of fl

In order
concepts, t
amples of
with comm
pedance le

† U. S. Ar

are for the purpose implied by the name and will naturally be employed to try to prove out the simulation and model results. Even during these field-test phases we are to utilize simulation techniques. In this context the 709 will be linked to that part of the system being subjected to evaluation in a field operational environment; a single thread employment of equipment will be supplemented by simulation of the remainder of the system. (See Fig. 2.) That is, the computer will provide the data sink to introduce input into the system and to absorb output from these echelons actually being operated.

The Computer Center became operational in February, 1959. The larger part of the application studies are completed, and detailed analysis has begun on several of them with demonstration runs already made on at least two. The major hardware items of the prototype system are on order, with the first to be delivered this coming Fall. Combining this progress with that reported by Captain Luebbert on hardware, transmission, and programming aids we remain confident that our objectives can be attained.

In summary, this paper has reported on an ambitious and futuristic program undertaken by the Signal Corps to provide the Army with a vast tactical Automatic Data Processing System. The proposed system in prototype form is to be operational by 1963 and will incorporate the very latest developments in digital techniques, *i.e.*, new miniaturized general-purpose data-processing devices, computer-to-computer communications, and automatic programming. The research efforts in this project, and certain standards derived, are bound to have an effect on and contribute to related commercial data-processing activities.

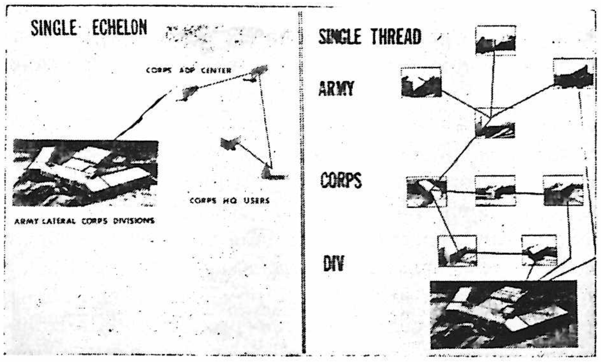


Fig. 2—The IBM 709 will serve as a source and destination of system input and output by simulating the missing echelons during field tests.

Data Transmission Equipment Concepts for FIELDATA

W. F. LUEBBERT†

FIELDATA is an integrated family of data processing and data transmission equipment being developed for Army use. A unique feature of this family is the almost complete disappearance of conventional distinctions between communications and data processing. This paper deals primarily with the concepts and techniques developed to create this evolutionary merger emphasizing the ways in which conventional communications concepts have been adapted to achieve a high degree of interoperability with computers and other data processing equipment, and an extraordinary degree of flexibility and adaptability of application.

In order to explain and illustrate the FIELDATA concepts, this paper makes extensive use of specific examples of design decisions, particularly those dealing with common features such as codes, voltage and impedance levels, data rates, etc. Among the equipments of the FIELDATA family developed in accordance with these concepts and common standards are the following, all of which are scheduled for completion prior to the end of 1960: the MOBIDIC computer (Sylvania), the BASICPAC and LOGICPAC computers (Philco), the AN/TSQ-35 19,200 bit/second data transmission equipment (Bendix-Pacific), the AN/TSQ-33 2400 bit/second data transmission equipment (Collins), the AN/TSQ-32 1200 bit/second data transmission equipment (Stelma), the DATA COORDINATOR, a facilities coordination and control equipment for an integrated communications and data processing system (IBM), and a host of miscellaneous equipments such as magnetic tape transports (Ampex), a flexowriter-like electric typewriter (Smith-Corona), high-speed printers (Anderson-Nichols), security equipment (Collins), etc.

The fundamental capabilities of data processing equipment can be described as the ability to transform

† U. S. Army Signal Res. and Dev. Lab., Fort Monmouth, N. J.

information into more desirable or useful forms. Using the same viewpoint, the fundamental capability of conventional communications equipment can be described as the ability to transfer information to a more desirable or useful location. Of course, from an abstract logical viewpoint a transfer is merely one special kind of transformation, one in which only physical location is changed. Combining these descriptions, one is led to the concept of a generalized data handling system capable of performing generalized transformations, of which conventional one-location data processing would be one special case and conventional no-processing data transmission would be another special case. In such a system there would be no fundamental distinctions between data processing and data transmission, but only distinctions of convenience based upon application, use, and design emphasis.

Is such a concept a reasonable one, and does it have any practical utility? The answer to the first part of this question will be examined in detail in this paper; the answer to the second part will be determined by the success in actual use of the equipment and system concepts it generates, the concepts explained in this paper which are receiving initial implementation in the FIELDATA family of equipments.

An examination of the information flow and manipulation in typical data processing and data transmission equipments shows almost immediately that there is a very considerable mixture of functions going on in both types of equipment. A very considerable amount of the activity going on inside any computer or data processor consists of simple transfers of information from one part of the processor to another. Within data processing equipment a major part of the activity is concerned with the generation, manipulation, and other processing of information used for control, supervisory, and error reduction purposes. In many cases nearly identical operations go on in both data processing and data transmission equipments, with the differences, if any, being matters of design emphasis based upon application and use.

Many practical cases of this similarity are immediately obvious, particularly in the area of devices used for data entry and output. For example, computers frequently use paper tape readers similar to those used in teletypewriter transmitter distributors, and paper tape punches that could easily be used in teletypewriter reperforators. Similarly the idea that kinds of input-output devices such as card readers and punches and magnetic tape transports which are widely used for data processing can effectively be adapted for use of communications lines is also being exploited in equipments such as the IBM transceiver, the Collins Kinetape, etc.

FIELDATA emphasizes this kind of exploitation to the extreme, particularly encouraging it by assuring that interconnection of semiautonomous equipment modules be made in accord with common standards without distinction whether these equipments are conceived primarily for computer-associated or transmission-associated functions. This makes it possible for the same data terminals to operate not only with data processing-type inputs and outputs such as computers, paper tape, magnetic tape, and IBM cards; but also for it to operate with real time weapons system data and with telegraphic data.

Control circuitry is so devised that pure binary as well as alphanumeric (*alpha*betic-*numeric*) data may be handled. Since any digital code, be it Baudot (teletypewriter) code, Holerith (IBM card) code, or any of a wide variety of computer codes may be represented in binary bit-by-bit form, the FIELDATA devices have the potential of transmitting or handling any type of digital data. Thus, they could be used with digitized voice, digitized facsimile, or other types of digitized analog signals.

The use of common standards, codes, and standard data rates makes possible the kind of data transmission equipment concept shown in Fig. 1. This concept leads naturally to a division of the subassemblies of data transmission equipment into three kinds:

1) *Input-output transducers* are devices for converting information from some human or machine usable form such as paper tape, magnetic tape, punched cards, analog electrical voltages, strokes on a keyboard, etc., into digital form.
2) *Transmission transducers* are devices for converting data in digital form into appropriate signals for transmission over radio, wire or other kinds of propagation media.
3) *Embolic*[1] *equipment*, normally inserted between input-output transducers and transmission transducers, is used primarily to perform control and supervisory functions, error detection and/or correction, buffering, and/or speed conversion, code conversion, or encryption necessary for proper system operation of the data transmission equipment. The functions of embolic equipment are information processing functions. Inputs and outputs will both be digital in form, although supplementary analog information may also be available particularly in some kinds of error control schemes. A general-purpose computer is potentially a very powerful and flexible type of embolic equipment, but the necessary functions can often be performed much more economically by specialized equipment.

This division of subassemblies may be a physical division into separate items of equipment, into semiautonomous parts of a single equipment (either in a separate box or in a single box) or it may be merely conceptual with no physical implementation.

---

[1] Embolic is a coined term from the Greek *embolisimos* meaning to put between or insert. This word is also used in medical, astronomic, and ecclesiastic literature to describe other specific kinds of intercalations.

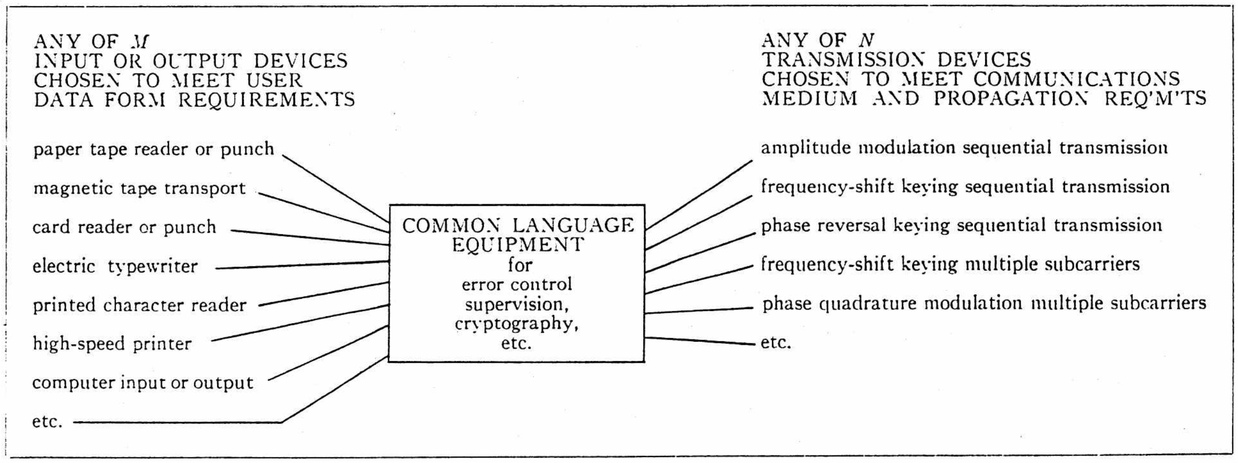

ANY OF *M*
INPUT OR OUTPUT DEVICES
CHOSEN TO MEET USER
DATA FORM REQUIREMENTS
paper tape reader or punch
magnetic tape transport
card reader or punch
electric typewriter
printed character reader
high-speed printer
computer input or output
etc.
COMMON LANGUAGE EQUIPMENT
for
error control
supervision,
cryptography,
etc.
ANY OF *N*
TRANSMISSION DEVICES
CHOSEN TO MEET COMMUNICATIONS
MEDIUM AND PROPAGATION REQ'M'TS
amplitude modulation sequential transmission
frequency-shift keying sequential transmission
phase reversal keying sequential transmission
frequency-shift keying multiple subcarriers
phase quadrature modulation multiple subcarriers
etc.

Fig. 1—FIELDATA equipment modularity concept.

Obviously maximum flexibility and adaptability can be achieved if any input-output transducer can operate with any transmission transducer. This permits one to tailor equipment to meet the requirements of a particular situation. It permits one to choose the in-out device suitable for the user's most convenient data form (paper tape, magnetic tape, IBM cards, etc.) independently of the nature of the transmission facility available. One can then choose the transmission transducer on the basis of the transmission medium used (VHF radio relay, HF radio, loaded cable, carrier transmission on wire, etc.) independently of the user's data form. Then join the two together taking advantage of common standards of interconnection or intercommunication between modules to create a well-tailored combination.

The question then arises, "Why embolic equipment?" Certainly by proper choice of common interconnection or intercommunications characteristics one can minimize requirements for code conversion, buffering, speed conversion, etc. Ideally one should be able to join the input-output transducer to the transmission transducer without embolic equipment, so why have it? The answer is that there are three important functions in a data terminal which seem convenient to separate from both in-out transducers and transmission transducers: 1) communications supervision, 2) error control, 3) cryptographic security.

Supervisory functions are conveniently separated from the input-output transducers and the transmission transducers because supervisory requirements may be strongly influenced by both. Error control requirements may also be strongly influenced by both the in-out data and the transmission situation, and may even in some cases not be desired at all. Thus it is best treated as a module which can be tailored to the paired requirements of in-out and transmission or completely omitted. Cryptographic security is conveniently a separate module so that it may be omitted in those cases where security is not required or desired.

In order to discuss supervisory activities conveniently it is desirable to make a distinction between two kinds of information which flow through a communications system:

1) *Primary information* is that which a user wishes transferred to another location, that is, the information to be communicated.
2) *Secondary information* is added to the primary information either by the originator or by equipment of the communications system which is used to perform functions of supervision, routing, error control and related activities necessary or desirable to permit the primary data to be effectively communicated. This information is used by communications equipment and personnel and is normally of no use or interest to the ultimate recipient of the primary information.

A basic requirement for maximum flexibility and adaptability is that the user, who enters data at an input device and receives data at an output device, need not be required to exercise judgment or knowledge, to perform special activities, to use or to interpret secondary information. This, in turn, makes it desirable to isolate secondary information from the input-output devices, which are the user's point of contact with the communications system. A convenient way of establishing and enforcing this isolation is the creation of a distinct embolic module which generates, receives, interprets, and/or acts upon secondary information and passes on action requirements derived from this secondary information over local control lines to its associated input-output transducer and transmission transducer. It is important to note that such equipment need not have any ability to interpret or act upon primary data, and thus its operation can be made completely independent of the coding used for primary data so long as there is a unique method of distinguishing primary from secondary data.

In the FIELDATA family of equipments the segregation of primary and secondary data is accomplished in a particularly simple way by providing separate transmission symbol "alphabets" for primary and secondary data. The basic unit of data in FIELDATA codes is a six-bit character. For primary data this may be merely a six-bit block with no specific meaning assigned to specific bits within it, and no restrictions on the permissible code combinations which may be transmitted. If this information is alphanumeric in form so that specific bit meanings must be assigned for electric typewriters, printers and similar input-output transducers to operate, then the FIELDATA alphanumeric code given in Appendix I is used. A similar but distinctly different six-bit FIELDATA supervisory data code has also been created and is outlined in Appendix II. Message heading and addresses, dialing information, multiplexing information, signals indicating start and end of message blocks, and control signals are all created using this alphabet.

In many situations, for example inside computers, it is desirable to both the six-bit primary data alphabet and the six-bit FIELDATA supervisory code alphabet with implicit differentiation between them similar to the implicit differentiation between data and instructions in the computer. However, other situations arise where it is desirable to provide an explicit identifying tag to specify which alphabet is being used.

A basic tagging method has been adopted for use on interconnecting cables employed for joining equipment modules. A seventh or tag bit is added to each six-bit symbol: a binary "one" if the symbol is primary data, and a binary "zero" if the symbol is from the FIELDATA supervisory alphabet. In the interconnecting cables an eighth bit in the form of an odd parity bit is also added to provide error protective redundancy. This basic eight-bit form creates the appearance of an eight-bit code. It has widespread use, but cases do exist where other alphabet tagging techniques and other redundancy is preferable. For example, serious complications would arise if this basic eight-bit form were used for the actual punching of FIELDATA information on paper tape. This occurs because control difficulties occur with a paper tape punch-reader system when primary data symbols are allowed to use either the blank tape condition (no channels punched) or the deleted tape condition (all channels punched). Since the use of the basic eight-bit form inevitably results in one or the other of these symbols being a primary data symbol depending upon whether a hole is interpreted as a binay "zero" or as a binary "one," a slightly different tagging and parity scheme must be adopted. This, of course, results in another eight-bit form which could be interpreted as an eight-bit code. Other tagging and redundancy schemes could lead to other apparent codes. Thus, there is no such thing as a *unique* eight-bit FIELDATA code, although it may be convenient to think of the basic eight-bit form as an eight-bit code.

In some situations transmission errors may create important reasons for using other tagging and error protective redundancy schemes. Errors which cause loss or change in supervisory information such as dialing information, message headings, start and end of message indications, and so forth, may completely disrupt the proper functioning of a communications system. Thus they may require protective redundancy many times more powerful than the simple parity used in the basic eight-bit form. However, compared to these secondary data, primary data may be capable of tolerating considerably higher error rates. For example, if the data are English text the inherent redundancy of the language may permit significant corruption without loss of intelligibility. Numerical data may consist of successive observations of a physical phenomenon in such a form that inconsistent data may be deleted and ignored, or they may be protected by numerical checks similar to those used by accounting systems to detect bookkeeping errors. In a situation where error rates were severe it might not be desirable to apply as powerful error control to these less demanding primary data as to secondary data; thus one might desire to use methods of differentiating between primary and secondary data which are more resistive to corruption by errors than the simple tags used in the basic eight-bit form together with different error control schemes for primary and secondary data.

This kind of differentiation illustrates a fundamental assumption about error control which is part of the FIELDATA concepts. Specific error control requirements should be determined by the nature of the data and their use. As mentioned above there may be cases where primary data can tolerate frequent errors with little loss of usefulness, but there are other cases, such as the transmission of computer programs, where very low error rates are required. In some cases if errors above the desired maximum rate occur and are detected, the erroneous information may be deleted without significant harm; in other cases they must be corrected. In some cases where correction is required it may be deferred and handled by a service message; in other cases it must be corrected before the data are released, and so forth.

Notice that all these requirements are determined by the use of the data, and that these demands remain the same regardless of the transmission path and types of transmission equipment through which the data may be required to flow. However, the occurrence of errors is anything but independent of transmission factors. Although errors do occur in input-output devices and embolic equipment, by far the most variable and difficult to control errors normally occur in the transmission portion of a data link. These errors are often quite variable in frequency and interrelated in occurrence. They are quite strongly dependent upon the nature of the propagation medium and the kinds of disturbances to which it is subject. For example, an HF radio link subject to severe fading and multipath disturbances could hardly be expected to have the same error problems as a wire and cable link subject to intracable crosstalk and impulse-type switching noise. Furthermore, for a given propagation medium and kind of noise and disturbances, the frequency and interrelationships of errors are strongly dependent upon the characteristics of the transmission transducers used. For example, if amplitude modulation is used one would expect different errors than if frequency shift keying were used; if sequential transmission of short bauds on a single subcarrier is used one would expect different error problems than if parallel transmission of long bauds on multiple subcarriers is used; and if sampling or nonintegrating types of detection are used one would expect different error problems than if full integrating detection schemes were used.

The number, variability, and difficulty of measurement and analysis of the various factors which contribute to the frequency and interrelationships of transmission errors or particular circuits is staggering. At the present time the state of the art is such that only crude estimates of frequency of error can be made for typical equipments when exposed to disturbances other than Gaussian noise, and that practically nothing can be estimated in advance about the interrelationships of errors under practical conditions of impulse noise, crosstalk, propagation variations, etc. This is particularly unfortunate because the effectiveness of the various digital error control schemes available is strongly dependent upon the interrelationships of errors. For example, a simple parity check is capable of detecting single errors but not double errors. If errors occur randomly, double errors will seldom occur and this very simple check will be quite powerful. Thus if the bit error rate is 10^{-4}, a simple parity check will reduce the undetected error rate to 10^{-8}. On the other hand, if errors tend to be clustered a parity check will be rather ineffective. Thus if the conditional probability of a second error immediately following the first is 0.5 and the bit error rate is 10^{-4}, then a parity check will reduce the undetected error rate to only 0.5×10^{-4}. Given knowledge of the interrelations, checks can be designed which give high protection with a minimum amount of checking equipment. Unfortunately this knowledge is usually unavailable.

If the most important sources of errors are in a data communications link, is it proper to incorporate the major error control features of the link into the transmission transducers? The FIELDATA concepts answer this question with a resounding NO! Why? The key reason is that while the *occurrence* of errors and the raw error rate and characteristics are determined primarily by transmission factors, the error *requirements* are determined by the use of the data. The means and techniques of error control appropriate to a particular situation obviously depend upon the *interaction* of these factors. However, it is a fundamental modularity principle of FIELDATA that the characteristics of transmission transducers should as nearly as possible be independent of the details of user input-output characteristics and data employment.

If one were to incorporate the error control features into the transmission transducers and one desired to provide p classes of controlled error service to users with q modulator/demodulator assemblies, then one would require p times q types of complete transmission transducers. The obvious answer is to separate modularly, making the error control module an item of embolic equipment. This allows one advantage of similarities in requirements and raw error characteristics among the different situations to reduce the variety of equipment to be constructed.

In view of the present difficulties in predetermining the specific error characteristics of transmission transducers prior to construction and test, it also permits construction of new transmission transducers and their use with existing error control embolic equipments until the specific error characteristics of the transducer can be measured and new error control embolic equipment designed if necessary to meet user requirements with the measured transmission error characteristics.

In addition to these practical advantages of placing error control responsibilities in embolic equipment modules rather than in transmission transducer modules, there are conceptual advantages associated with maintaining the simplest possible information flow patterns and division of activities among the three basic kinds of assemblies.

In general transmission, transducers pay no attention to the information content of the digital information they convert to modulated transmission form, neither knowing nor caring whether the data are primary or secondary, whether they are redundant or irredundant, or what code or codes they use. In contrast to this, embolic equipments normally act as information processing devices. Thus, in supervising a transmission link embolic supervisory equipments act on sensory information received from transmission transducers and in-out transducers and generate, process, and interpret secondary supervisory information to control the over-all operation of the communications link. The error control problem is exactly parallel. Acting on information about user requirements from the in-out transducer side, and information and sensory information about transmission errors from the transmission transducer side, error control equipment is required to generate, process, and interpret error control information using it to control (often via supervisory operations) the overall operation of the communications link in such a way as to control its errors. Thus it is obvious that from the information processing viewpoint the performance of error control as an embolic function has a close parallel to other embolic functions and is distinctly different from the functions otherwise performed by transmission transducers.

Some of the FIELDATA equipments being developed in implementation of these concepts are listed in Table I.

The transmission transducers of the FIELDATA family are limited in number. However, the choice of the 75×2^n pattern of data rates permits widespread augmentation by minor modification of existing or developmental teletypewriter multiplexed transmission equipments. In addition, future expansion is simplified by the fact that future equipments will be able to utilize the same embolic and input-output equipments, and will thus be cheaper to develop than data transmission equipments which require development of embolic and/or input-output equipment as part of the same package. Expected future expansion will place greater emphasis on transducers for radio circuits.

TABLE I

Transmission Transducers		
AN/TSQ-32	600–1200 bit/second	Stelma Corp.
AN/TSQ-33	600–2400 bit/second	Collins Radio
AN/TSQ-35	19,200 bit/second	Bendix-Pacific
General Purpose Computers		
MOBIDIC		Sylvania
BASICPAC/LOGICPAC		Philco
INFORMER/DATA COORDINATOR		IBM
In-Out Transducers		
Electric Typewriter		Smith-Corona
Paper Tape Reader		Smith-Corona
Paper Tape Punch		Smith-Corona
High-Speed Printer		Anderson Nichols
Paper Tape Transport		Ampex Corp.
Magnetic Tape Transport		Ampex Corp.
Tacden		Aeronutronics Systems
Special Embolic Equipment		
CV-689	Cryptosecurity Adaptor	Collins Radio
CV-690	Control Equipment	Collins Radio
CV-691	Data Concentrator	Collins Radio

The AN/TSQ-32 is a transmission transducer capable of accepting data from a standard FIELDATA connection and of transmitting it serially as frequency-shift modulation of a single subcarrier over a standard voice channel. Transmission rates are 1200 bits/second (1500 words per minute) over good quality circuits, and 600 bits/second over poorer circuits.

The AN/TSQ-33 is a transmission transducer capable of accepting data from a standard FIELDATA connection and of transmitting it as 8 channels as synchronous phase quadrature modulation of four subcarriers over a standard voice channel. Transmission rates are 2400 bits/second (3000 words per minute) over good quality circuits, with 1200 and 600 bits/second available for use over poorer circuits. This equipment is essentially a militarized, miniaturized, FIELDATA compatible version of the Kineplex TE-206.

The AN/TSQ-35 is a transmission transducer capable of accepting data from a standard FIELDATA connection and of transmitting it by amplitude modulation of 8 subcarriers orthogonally spaced in frequency and time. The transmission baseband is the 48-kc band between 12 and 60 kc used for military and commercial cable and carrier circuits. Representative circuits are the Army spiral-four and associated AN/TCC-7, 8, and 11 equipments; and commercial types N and K carrier equipments. In addition to point-to-point full duplex operation, this equipment has special features for multiple station "common net" round-robin type operation.

In addition to these transmission transducers specifically designed for FIELDATA a wide variety of existing military and commercial teletypewriter transmission equipments can be used for FIELDATA transmission with only minor modification. Examples of such equipment are the AN/FGC-54 capable of transmission and diversity reception of FIELDATA information at 2400 bits/second over long-haul 3-kc radio circuits using 32 channels each operating at 75 bits/second. Another example is the AN/FGC-29 potentially capable of transmission and diversity reception FIELDATA information at 1200 bits/second using 16 channels each operating at 75 bits/second. Yet another example is the AN/TCC-30 potentially capable of transmitting 1200 bits/second using 16 channels operating at 75 bits/second.

The FIELDATA computers form the most complete and well-balanced portion of the FIELDATA family.

The FIELDATA computers may act as either input-output transducers or as embolic equipments for data transmission purposes, having the ability to accept, process, and emit either primary or secondary data. All are designed for direct operation with data transmission equipment. In their employment the MOBIDIC, which was the first machine to have its characteristics frozen, is the least capable of transmission of machine data; it has serious restrictions on its use of the supervisory code functions. The most flexible in its employment of data transmission will be the DATA COORDINATOR, a newer equipment which will have the capability of terminating a large number of data transmission circuits simultaneously and which will have a number of special capabilities and console positions to facilitate its use as a facilities coordination processor for an integrated communications/data processing system.

All the FIELDATA computers are general-purpose processors of modular design and great flexibility. Their relative computation speeds are shown on the bar chart. All are designed for field use with operation and maintenance simple enough for field personnel. The largest, MOBIDIC, mounts in a semitrailer van, while the others mount in shelters which can be carried on a truck. It is interesting to note that the minimum assembly which forms a fully operational stored program computer of the BASICPAC/LOGICPAC type or the INFORMER/DATA COORDINATOR type weighs in either case less than 150 pounds subdividable into 50-pound or smaller packages. However, these central processors are normally used for data processing purposes in vehicular assemblies which far exceed these weights because of auxiliary equipments such as multiple tape transports, special real-time input-output units, additional magnetic core memory modules, data transmission equipment, input-output converters, etc. For example, a vehicular BASICPAC would augment the central processor with four magnetic tape transports, a high-speed paper tape reader and punch, an AN/TSQ-33 transmission transducer, and other auxiliary equipment. Fig. 2 indicates the relative computational capabilities of the various FIELDATA processors.

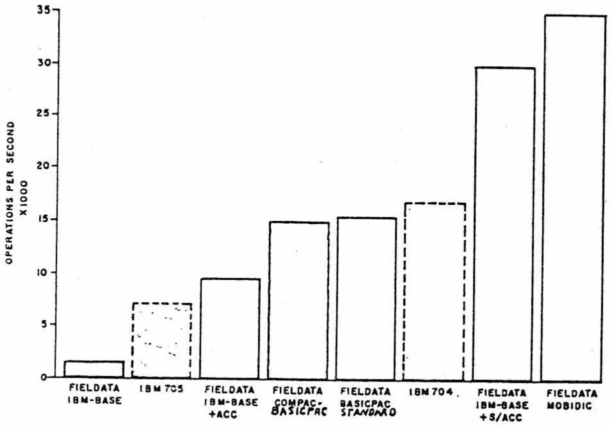


Fig. 2—Speeds of computation, FIELDATA processors.

The input-output transducers of the FIELDATA family serve double duty as computer input-output devices and as data transmission input-output devices. The group under current development constitutes a minimum group of general usage items, a number of which are in only partially militarized form. This minimum group will be augmented by future field teletypewriter equipments which will utilize FIELDATA code, and by advanced equipments now under study to provide specialized input-output capabilities. Since the items are mostly quite conventional, detailed descriptions are omitted here in order to save space.

The specialized embolic equipments in FIELDATA provide cryptographic security, interconnection of input-output and transmission transducers, and tie-in of FIELDATA circuits to existing teletypewriter circuits.

Three major items of special embolic equipment are being developed for FIELDATA. The CV-689 is a special cryptosecurity adaptor which permits an existing type of security equipment to be inserted just before the transmission transducer in any FIELDATA data transmission assembly, thus providing cryptographic security.

The CV-690 is a device which provides supervisory control, error control, and synchronizing buffer facilities for connecting paper tape or magnetic tape units to the AN/TSQ-32 or 33. Although future plans call for similar special militarized embolic equipments for other kinds of input-output equipment such as card equipment none is now under development. However, rather minor modifications of commercial Collins Kinecard equipment will permit nontactical employment of AN/TSQ-33 equipment for card transmission, and at least some versions of the MOBIDIC will include card equipment which, through the computer, can reach transmission facilities.

Although FIELDATA concepts make provision for a wide variety of error control and supervisory control systems only the particular system used in the CV-690 will actually be used initially except for computer-to-computer transmission, since other embolic devices using different error control or supervisory control schemes are now under development. It is expected that when experiments determine the actual frequency and interrelationship of errors for particular transmission transducers, more effective schemes will be devised. However, the very simple and easy to implement two-dimensional (interlaced) parity error detection scheme followed by request-back or rerun-type error correction used in the CV-690 is particularly easy to implement, and is expected to suffice for initial testing.

The CV-691 Data Concentrator is a device designed to bridge the gap between existing large-scale 60- and 100-wpm teletypewriter facilities and FIELDATA equipment. Although all FIELDATA processors have normal provision for a paper tape reader which accepts teletypewriter tape as well as one for FIELDATA tape, there exists a significant need, especially at locations where FIELDATA processors might not be available, to accept multiple channels of teletypewriter information and convert it into FIELDATA form to take advantage of FIELDATA transmission transducers and error control, permit recording on FIELDATA magnetic or paper tape, permit printing on FIELDATA high-speed printers, simplify entry into FIELDATA computers, etc. The CV-691 accepts up to 25 (or 50) teletypewriter inputs, stores the information in a buffer core memory (made up of the same memory planes as used in MOBIDIC), assembles it into message blocks, converts it into FIELDATA form, applies the same error and supervisory control as the CV-690, and emits the data in FIELDATA form at rates up to 2400 bits/second (3000 words per minute). The receive side performs the inverse functions.

It is expected that as more and more of the voice and other analog communications systems convert to pulse code modulation and other digital forms that additional types of FIELDATA embolic equipment will be required to perform the error control, supervisory functions, and buffering/synchronization necessary to tie input-output transducers to their digital bit streams.

The FIELDATA family is an attempt to create an integrated family of data transmission equipments to meet Army needs. Though lacking many of the features and equipments of an ideal family of data transmission equipments, it will make available in experimental quantities by the end of 1960 the first integrated family of equipments for experimental establishment of a truly integrated communications/data processing system. It will be the first system in which data processing and communications equipment both utilize the same input-output and storage devices, the same voltages, impedance levels, codes, and other common interconnection characteristics, and in which the equipments are so designed that in many cases the only way to determine whether a device is used for communications or data processing is to look at its specific application in the system.

APPENDIX I

FIELDATA ALPHANUMERIC CODE

A key step in achieving the required compatibility between the various elements of an automatic data system is the adoption of a common "language" for the storage and transmission of data throughout the system. The basic 6-bit alphanumeric code for use in this family shall consist of 2 indicator bits and 4 detail bits. The pattern of character assignment for the code is as follows with the 2 indicator bits determining the choice of column and the 4 detail bits determining choice of row in Table II.

APPENDIX II

FIELDATA SUPERVISORY CODE

The FIELDATA supervisory code is used for message headings, dialing, multiplex identification, supervisory control, and other activities associated with secondary data. This code is similar to the FIELDATA alphanumeric code used for primary data. It also consists of 2 indicator bits and 4 detail bits. The pattern of control assignment is as follows with the 2 indicator bits determining the choice of column and the 4 detail bits determining the choice of row in Table III. When it is not necessary to provide alphabetic supervisory information only the latter two columns are used. In this case when the basic 8-bit FIELDATA form is used, an OR of the first indicator bit and the tag bit will provide clocking for the 96 legitimate characters of the 8-bit form.

TABLE II

	00 (Upper and Lower Case)	01 (Upper and Lower Case)	10 Upper Case	11 Lower Case
0000	Master Space ≪	K	)	0
0001	Upper Case ʒ	L	–	1
0010	Lower Case %	M	+	2
0011	Tab. ◇	N	<	3
0100	Car. Ret. <	O	=	4
0101	Space △	P	>	5
0110	A	Q	–	6
0111	B	R	$	7
1000	C	S	*	8
1001	D	T	(	9
1010	E	U	"	'
1011	F	V	:	;
1100	G	W	?	/
1101	H	X	!	.
1110	I	Y	,	Special □
1111	J	Z	Stop ⊕.	Back Space

TABLE III

	00	01	10	11
0000	BLANK/IDLE	k	Dial 0	Ready to Transmit
0001	Control Upper Case	l	Dial 1	Ready to Receive
0010	Control Lower Case	m	Dial 2	Not Ready to Receive
0011	Control Tab.	n	Dial 3	End of Blockette
0100	Control Carriage Ret.	o	Dial 4	End of Block
0101	Control Space	p	Dial 5	End of File
0110	a	q	Dial 6	End of Control Block
0111	b	r	Dial 7	Acknowledge Receipt
1000	c	s	Dial 8	Repeat Block
1001	d	t	Dial 9	Spare
1010	e	u	Start of Control Block	Interpret Sign
1011	f	v	Start of Block	Non-Interpret Sign
1100	g	w	Spare	Control Word Follows
1101	h	x	Spare	S.A.C.
1110	i	y	Spare	Special Character
1111	j	z	Spare	Delete

CACM v2 #5 May 1959

problems, information retrieval, foreign language translation, etc.

Manuscripts are invited for all four categories. The committee responsible for the newsletter is: Steven G. Vandenberg, Mental Health Research Institute, The University of Michigan; Bert F. Green, Lincoln Laboratories, Massachusetts Institute of Technology; Bruce MacKenzie, International Business Machines; John Mauchley, Remington Rand Univac; Martin Rogoff, U. S. Bureau of Mines; and Charles Wrigley, Michigan State University.

The level of presentation we will strive for is in between the elegance and length of the *Journal of the Association for Computing Machinery* and the more introductory treatment of topics and brevity of commercial news journals such as *Datamation* or *Computers and Automation.*

The newsletter will be published as a department in the quarterly journal, *Behavioral Science*, which is published by the Mental Health Research Institute of The University of Michigan. The subscription price of *Behavioral Science* is $6.00 per year, or $5.00 per year to persons who identify themselves as members of a recognized national professional or scientific organization. We would like to invite departments, computer installations, and libraries, as well as interested individuals, to subscribe.

Will you please bring this newsletter to the attention of your staff and students? Thank you very much.

Steven G. Vandenberg
Mental Health Research Institute
The University of Michigan
Ann Arbor, Mich.

To the Editor:

There have been some recent comments in the *Communications* on the problem of finding square roots on a machine which cannot divide easily. This calls to mind the following simple technique:

To find $\sqrt{N}$, we first find $1/\sqrt{N}$. This is done by solving the equation $f(x) = N - 1/x^2 = 0$ by Newton's method. The resulting iteration is $x_{n+1} = .5x_n(3 - Nx_n^2)$, which requires no divisions. Having found $x = 1/\sqrt{N}$, one more multiplication completes the process: $\sqrt{N} = Nx$.

David Morrison
Space Technology Laboratories
Los Angeles 45, Calif.

To the Editor:

The following 64-character alphabet (Table 1) is suggested as a possible point of departure for hardware implementation of the Reference Language being developed by the participants in the ACM-GAMM Conference on an International Algebraic Language.[1, 2] Inherent in this scheme is a one-for-one correspondence between representations on the written or printed page (the link from and to humans), and in any hardware-readable data medium (the link with computing machines: punched cards, punched tape, magnetic tape, etc.). Control characters provide, in addition to hardware control, for the vital requirement that the alphabet be extensible.

The Reference Language was intended to be "th

TABLE 1

64-Character Alphabet

6-Bit Code	Character (Shape)	Meaning	6-Bit Code	Character (Shape)	Meaning
00	0	zero	40	—	minus
01	1	one	41	J	
02	2		42	K	
03	3		43	L	
04	4		44	M	
05	5		45	N	
06	6		46	Ɵ	letter "Ɵ"
07	7		47	P	
10	8		50	Q	
11	9		51	R	
12	⌣	or	52	↓	exponentiation (lower)
13	=	equal to			
14	'	single quote	53	$	basis of ten, or dollar sign
15	≡	equivalent to			
16	⌢	and	54	*	Multiply by
17	;	statement separator	55	<	less than
			56	[	subscript bracket (left)
20	+	plus	57	:	label separator
21	A				
22	B		60	ƀ	blank space
23	C		61	/	divide by
24	D		62	S	
25	E		63	T	
26	F		64	U	
27	G		65	V	
			66	W	
30	H		67	X	
31	I				
32	e	end of line, or end of card, or carriage return	70	Y	
			71	Z	
			72	d	DOWN SHIFT
			73	,	list separator (comma)
33		radix point			
34	)	parenthesis (right)	74	(	parenthesis (left)
35	↑	exponentiation (raise)	75	>	greater than
			76	¬	not
36	]	subscript bracket (right)	77	n	null
37	u	UP SHIFT			

---

[1] Perlis, A. J., and Samelson, K. for the ACM-GAMM Committee, Preliminary report—international algebraic language, *ACM Communications 1*, No. 12, 8–22 (Dec. 1958).

[2] ACM-GAMM Committee, Report on the algorithmic language ALGOL, *Numerische Math. 1* (Jan. 1959).

guide for transliteration between Publication Language and various Hardware Representations." The purposes of the present proposal are to show that an essentially explicit form of the Reference Language itself can be written in a six-bit-per-character alphabet, and to urge development and hardware realization of one such alphabet for digital computer input and output.

The suggested alphabet provides for unambiguous representation of lower-case letters and does not use overprinting. It includes all ten decimal digits with unique "zero" and "one"; all twenty-six letters of the English alphabet, with unique "I" and "Ꝋ"; twenty-three other printing characters; four control characters, two of which are always printed; and "blank" (which is the same as "space").

The fact that the required 64 characters can be represented explicitly by a six-bit code permits straightforward adaptation of present-day computer data input-output media. Among these are six-active-track tape (paper as well as magnetic), twelve-row punched cards, and twelve-active-track magnetic tape. CR tube printers, several high-speed mechanical printers, and conventional electric typewriters and typewriter-reader-punches can print the required number of character shapes.

Essentially all present-day computer data input media, except IBM-Hollerith-coded punched cards, represent data by binary-coded characters. The most widely used arrangement designates each character by six data bits, frequently with one additional bit per character used for checking purposes.

IBM Card Code

The IBM card code offers simplicity of mechanical decoding in line printers and other heavy equipment for punched-card processing. In a typical computer-input application, the twelve-bit card-column representation is first converted to a binary or other reasonably compact representation, either by hardware (external or internal) or by program.

The IBM card has been popular as an input medium for large computers, presumably because of the ready availability of satisfactory equipment for punching, interpreting (printing card contents onto the card itself), listing (printing card contents on paper), reproducing (copying cards), and the performing of various logical operations with or from cards, and because of the great ease with which small items may be removed, changed, or inserted into an array.

The alphabet herein proposed could be represented in punched cards by the defining of sixteen new combinations of holes in addition to the forty-seven (plus blank) which constitute the standard IBM card code, together with the renaming of five of the existing combinations of holes. There is, however, no basic reason why the punched-card operations really necessary in computer applications (punching, listing, comparing, and reproducing) could not be performed by punching IBM cards with a six-hole code. Indeed, as pointed out by C. W. Bachman in 1958 in an unpublished communication to the members of SHARE, such a representation would permit the punching of two six-bit characters into each column, thereby increasing card capacity from 80 to 160 characters. The card code suggested here uses six bits per character, and may be punched either 80 or 160 characters per card.[3]

TABLE 2

Two-Character Symbols

The Two 6-Bit Codes	The (Two-Character) Shape	Meanings
75,13 or 76,55	>= or ¬<	Not less than (≧)
13,55 or 76,75	=< or ¬>	Not greater than (≦)
57,13	:=	Assignment delimiter (:=)
13,57	=:	Procedure delimiter (=:)
76,13	¬=	Not equal to (≠)
40,75	->	Substitution separator
37,72	ud	STOP CODE (paper tape only)

Control Characters and "Blank"

The proposed alphabet includes four characters which have control, rather than data, significance, and "blank." As input and output, these are represented by script letters as follows:

(a) d (DOWN SHIFT) is always written on coding sheets, punched into cards or tape, and is always printed explicitly, never implied; u (UP SHIFT) is implied to be in effect at the start of a card, line, or tape block, and is otherwise written, punched, and printed.

(b) e (END OF LINE) need not be written on coding sheets, although in coding sheets which are to be card-keypunched, at least the column corresponding to the last character in a card must not be used otherwise; it is always punched into cards; it is not printed when treated as alphanumeric data, but e or 011010 is printed when treated as binary data.

(c) n (NULL) may be punched over any character in order to delete it, i.e., to cause it to be ignored as alphanumeric input data; as binary input data, it is treated as the bit group 111111. n may be written on coding sheets to denote a null character. As alphanumeric printout it is ignored; as binary output it may be punched or printed either as n or as 111111.

(d) "blank" or "space" may for emphasis or clarity be written b on coding sheets or may be represented by no writing; it is always represented as 110000 in computer input media, except that in cards it need not be punched following an e. It is not printed, but is represented by a blank space, when handled as alphanumeric data, but is printed as b or 110000 when handled as binary data.

DOWN SHIFT and UP SHIFT provide for extensibility of the alphabet. Both are assumed to have a scope indefinite in extent (i.e., to "lock") and may be stacked or nested

---

[3] Note that the punching of control characters into cards reduces card capacity for data characters by at least one character: e is always punched, and several others may be punched.

(i.e., an UP SHIFT from the UP SHIFTed condition implies an additional shift level).

In the case of typewriters having shifting capability, d and u may also provide the additional clarity, for the twenty-six English letters only, of lower-case as well as upper-case characters.[4] Extra-shift designation is not explicit; e.g., after a DOWN SHIFT from the DOWN SHIFTed condition, such a typewriter will continue to type lower-case letters, but the significance to a computer program is that a one-level-lower case has been entered. This concept extends to yet-lower cases, and also to higher cases above u.

For the many types of printers having a working limit of 64 type faces, the u-d control function must be implied from the printed page, in a manner conceptually identical to that of parentheses which may be nested.

It is desirable that a reference convention be adopted for "normal" shift condition: I suggest that UP SHIFTed be assumed to be in effect at the start of a card, line, or tape block, so that absence of any shift characters will imply that all English letters are to be considered upper-case.

For input and output equipment, the minimum number of carriage control characters necessary seems to be one, which may be interpreted as END OF LINE, END OF CARD, or CARRIAGE RETURN, depending upon the hardware concerned. Under normal conditions, the e function is implied: the programmer starts a new line on his coding sheet; either the keypunch operator strikes a RELEASE key after punching the last character in or before column 79, or the machine senses that column 79 has been punched, and in so doing causes the e character to be punched; the computer and printer sense e as the end of a line of input listing or output printing. Without this character it would be necessary to single-space out to the end of a fixed-length type line or card, which would be quite costly in time. It is not clear that the two other carriage-movement controls commonly provided with typewriters (TAB, BACKSPACE), and such special controls as STOP CODE, are of sufficient value, or are generally enough applicable to other equipment, to merit single character alphabet representation. The STOP CODE function is needed for local control of certain classes of input equipment when operating in alphanumeric mode, i.e., paper tape readers. For this control function the two-character symbol ud is proposed.[5]

The punched-card representation of the character "blank" or "space" poses some special problems, since in the preferred encoding scheme the decimal digit "0" (zero) is designated by no punching. (On paper or magnetic tape, of course, blank-vs-zero distinction is not usually a problem. There is typically at least one checking bit (which for odd parity is a one for an all-zeroes character) and perhaps also a sprocket bit, for each character.)

The convention chosen is to punch, manually or automatically, a character (110000) for every "blank" character location on the card, until the end-of-card symbol e is punched. After the e, when handled as decimal data, "blank" is inferred at each character location, although before the e each "no punch" would mean "0" (zero). As binary input, the card contents are handled verbatim.

The other control character suggested here is designated NULL, n. Various computers have recognized control characters having certain null characteristics: in some cases, as in that of a conventional tape-controlled electric typewriter when used to read tape, such a character is passed over and not read; in others, as in output from some large-scale computers, a null character in an array of output data is simply ignored, i.e., it is treated as neither zero nor space, but causes the character before it and the character after it to be treated as though they had been adjacent in storage. It appears to be of general value to have a true null character for both input and output: as alphanumeric input or output, such a character would be ignored, but as binary input or output, it would be handled verbatim, as would any other array of six bits. In many cases when it is to be output from a computer, the null character will be constructed internally rather than being input as a special case.

Combined Symbols

The Reference Language representation as developed at Zurich included a certain amount of redundancy: > as well as <, and ≧ as well as ≦ were included. It is felt by many people that the convenience of either-sense inequality designation is significant. As a reasonable compromise, the meanings of all four symbols are provided in the proposed alphabet in this way: both > and < symbols are included, and the combining convention is chosen that ≧ may be represented by >= and ≦ by =<. The assignment delimiter := and the procedure delimiter =: are represented by juxtaposing the two symbols : and =. The use of ¬= to represent ≠ is certainly consistent; other logical operator combinatorial forms would seem to be suggested by this combination

---

[4] I am indebted to Mr. John Bartelt, of IBM-SEPD, for this interesting observation: In a typewriter having case shift capability, there need be no direct correspondence between (a) input or output electrical or mechanical signal, or punched tape or card character representation, and (b) type slug arrangement.

For instance, a typewriter may be able to print 88 characters from 44 two-symbol type slugs. Fifty-two of the 88 printable characters may be the 26 upper- and lower-case English letters; the identity of any letter may be specified by one of 26 bit patterns, while the choice of upper- or lower-case is specified by the shift condition at the time the character-print action is executed. Thirty-six of the remaining 38 bit patterns in a 6-bit code may represent explicitly as many as 36 printable characters on the other 18 double type slugs. For these, the bit pattern specifies shift condition as well as type bar identity; upon receipt of a signal the typewriter makes a case shift if necessary before or while the type bar motion is in progress. After printing that character, the typewriter is returned automatically to its previous case shift condition if that had been altered.

[5] Suggested to me by W. A. Ramshaw of United Aircraft Cor

Substitution Separator

The symbol "→" was adopted at Zurich, although many Americans prefer "←". Either can be constructed[5] by character pairs provided here, –> or <– respectively; the former is listed. Since a blank space on each side is required for visual clarity, these combining conventions really require four characters. This is not felt to be objectionable, since <– occurs only once per statement. Note that other connectives such as => and <=> may be formed readily as required outside the present IAL.

Decimal Scale Factor Designator

The subscript "$_{10}$" was used at Zurich to denote decimal scale factor. While the connotation is not exact, the symbol $ is here assigned, for use within this Reference Language representation only, to indicate a decimal scale factor. All with whom I have discussed this matter agree that other applications of the symbol $ are so important as to demand its presence; none object to the convention specified in the previous sentence. This use of $ is suggested for program source languages only.

Multiplication Designator

The Zurich notation used the multiplication cross ×. There is an obvious nuisance in the ambiguity with letter "X"; the representation of multiplication symbol by the asterisk * is well-established in the computer art and offers no ambiguity; the asterisk also has other valuable uses. Its use in the proposed hardware alphabet is felt to be non-controversial among Americans, although some of the Europeans objected to its use in the Reference Language proper.

Notes on Shapes of Characters, as Written and Printed

This seems to be an appropriate occasion for removal of certain minor ambiguities in written and printed character shape. Obviously, these are matters of personal taste or habit; I feel that some improvement should be made, and offer these suggestions pending receipt of better ones:

(a) Numeral "one" uses the exaggerated headstroke of the German script "1". Letter "I" uses exaggerated top and bottom bars.

(b) Letter "ϴ" includes a flourish at the top as when written as an English script capital letter. I feel that the presently common use of "Ø" to represent the letter "O" is confusing to the novice and is more laborious to write than "ϴ".

(c) The Boolean connectives "and" (cap ⌐), "or" (cup ⌙), and "not" (inverted check mark ¬) use right-angle corners for ease of writing and reading manuscript. This avoids ambiguities which may arise from either ⌣, ⌢ or ﹀, ︿, and seems consistent with the "¬" notation.

Suggested 6-Bit Encoding of 64-Character Alphabet

The scheme (Table 1) represents, as two octal characters, the binary codes for the alphabet described above. Forty-eight of these characters are identical with the 48 SHARE Standard Characters for internal representation in the 709 Computer. (The SHARE 709 character set differs from the SHARE 704 set only in that in the 709 set octal 14 denotes *single quote* ('), while in the 704 set both 14 and 40 denote *minus* (–).) The decimals and letters are conveniently arrayed for sorting and comparison, in that 0–9 and A–Z representations increase monotonically. Representations for the sixteen characters which make up the rest of the set have been chosen from various sources. As noted above, the same six-bit binary-coded representation is intended for all media: punched cards, punched tape, magnetic tape (odd parity for both punched and magnetic tape).

Remark

The exact shapes and significance of individual characters in an alphabet can hardly be "standardized" except for a particular class of problems in a particular environment. It is hoped that the alphabet described above and listed in table 1 demonstrates that a set of 64 characters can provide a reasonable Hardware Representation of the IAL Reference Language.

H. S. BRIGHT
Westinghouse Bettis
Atomic Power Division
Pittsburgh 30, Pa.

Recd 9/21/57

B

E. I. A Engineering Department Report of Meeting of the Sub-committee on Storage Media and Language, TR-24.4.1

30 June 1959 Washington, D. C.

Members Present

C. Toeneboehn, Chairman	Emerson Electric Manufacturing Co.
J. K. Nelson	Western Union
E. L. Myracle	General ELectric
E. H. Bosman	IBM
Harry Hayman	Federal Aviation Agency
J. H. Sweer	RCA
G. E. Williams	Electrodata Division, Burroughs
Miles V. Hayes	Jones & Lamson
Alfred Z. Purzycki	Teletype Corp.
H. Tholstrup	Commerical Controls Corp.
G. E. Lyons	IBM
E. G. Newman	IBM

1. The chairman called the meeting to order at 10:00 A.M.

2. The chairman submitted a proposal for a standard language. During committee discussion changes were made in it, and it was agreed that it would be attached to the minutes of the meeting.

3. Mr. Hayman agreed to make a study of the important Binary Codes. It is attached to the minutes of this meeting.

4. It was agreed that further discussion would be necessary and no definite action would be taken by the committee until the next meeting.

Respectfully submitted,

Harry Hayman

Harry Hayman

NMAH 310/11

EIA LANGUAGE STANDARD PROPOSAL
FOR EQUIPMENT UTILIZING ONE INCH 8 TRACK PERFORATED TAPE

1.0 Scope

This proposed standard for 8 track perofrated tape is intended to serve as a guide in the coordination of equipment design for compatibility between equipments produced by various manufacturers.

1.1 This standard applies to equipment input and output information only and is not intended to constrain internal language or coding.

1.2 This language standard applies to codes composed of hole combinations normal to the direction of tape travel.

1.3 Additional standards covering other storage media will be issued as they are developed.

2.0 Storage media used shall be 1 inch perforated tape in accordance with EIA standards proposal S.P. 588.

3.0 Track Numbering

The eight tracks shall be numbered in accordance with EIA standards proposal SP 601.

4.0 Direction of Tape Travel

Direction of tape travel is immaterial since these codes are stored in a single row perpendicular to the direction of tape travel.

5.0 Variations of this Standard.

This standard is inflexible for the alphanumerical portion of the code but does allow variations in translation from code to punctuation marks and symbols to permit use of existing equipment. It is a goal that these variations can be minimized by selection of the preferred codes for future designs.

6.0 Language

The interpretation of the various hole combinations possible for various usage shall be as shown in Table I.

BINARY CODE

TRACKS 87654321	All Uses	Communication	Computation	Numerical Machine Tool	Preferred
01100001	A				
01100010	B				
01110011	C				
01100100	D				
01110101	E				
01110110	F				
01100111	G				
01101000	H				
01101001	I				
01010001	J				
01010010	K				
01000011	L				
01010100	M				
01000101	N				
01000110	O				
01010111	P				
01011000	Q				
01001001	R				
00110010	S				
00100011	T				
00110100	U				
00100101	V				
00100110	W				
00110111	X				

	0	1	2	3
0	EOL	0	—	&
1	1	/	J	A
2	2	S	K	B
3	3	T	L	C
4	4	U	M	D
5	5	V	N	E
6	6	W	O	F
7	7	X	P	G
8	8	'	Q	H
9	9	@	R	I
10		BS		LC
11	EOR	,	.	
12		TAB		UC
13	P17	P14	P16	P15
14			CR	SP2
15				DEL

BINARY CODE (cont'd)

TRACKS 87654321	All Uses	Communication	Computation	Numerical Machine Tool	Preferred
00111000	Y				
00101001	Z				
00000001	1				
00000010	2				
00010011	3				
00000100	4				
00010101	5				
00010110	6				
00000111	7				
00001000	8				
00011001	9				
00100000	0				
00111011	,				
01101011	.				
00110001	/				
00001011		Stop	#	End Of record	
00101010		PI 3	PI 3	back space	
00111110		Tab	Skip	Tab	
01000000		—	—	Minus	
01110000		&		Plus	
01111010		lower case	SPI	lower case	
01111100		upper case	●	upper case	
01111111		tape feed	tape feed	Delete	

BINARY CODE (cont'd)

TRACKS 8 7 6 5 4 3 2 1	All Uses	Communication	Computation	Numerical Machine Tool	Preferred
1 0 0 0 0 0 0 0		car. ret.	End line	C.R. E. of B.	
0 1 1 0 1 1 0 1		PI5	PI5		
0 1 0 1 1 1 0 1		PI6	PI6		
0 0 1 1 1 1 0 1		PI4	PI4		
0 0 0 0 1 1 0 1		PI7	PI7		
0 1 1 0 1 1 1 0		punch 2 on	S.P.2		
0 1 0 1 1 1 1 0		Spec.	car. ret.		
0 1 0 0 1 0 1 0		PI2	PI2		
0 1 0 1 1 0 1 1		$	$		
0 1 0 0 1 1 1 1		error	error		
0 0 1 0 1 1 1 1		punch off	end card 2		
0 0 0 1 0 0 0 0	space				
0 0 1 0 1 1 0 0		print restr.	%		
0 0 0 1 1 0 1 0		PI1	PI1		
0 0 0 1 1 1 0 0		non print	@		
0 1 0 0 1 1 0 0		punch 1 on	*		
0 0 0 0 1 1 1 0			end card 1		
0 0 0 1 1 1 1 1			correction		

Minutes Of Meeting
Of The TR24.4 Subcommittee
On Language And Media

September 14 & 15, 1959 Atlantic City, N. J.

National Aviation Facilities
Experimentation Center

Members Present

Henry Tholstrup	Commercial Controls Corp.
Harry Hayman	Federal Aviation Agency
David Robins	IBM, Poughkeepsie
George E. Lyons	IBM, Poughkeepsie
Miles V. Hayes	Jones & Lamson Machine Co.
Julius Agin	R. C. A.
E.H. Bosman	IBM Endicott
J.H. Sween	R. C. A.
J.K. Nelson	Western Union
John S. Copeland	Fairchild Graphic Equipment
Gordon Burns	Teletype
H. Tholstrup	Commercial Controls Corp.
L.E. Knees	Reeves Soundcraft Corp.
Timothy McGuire	R. C. A. Communications

1. Meeting called to order at 9:30 a.m. by the Chairman, Mr. H. Tholstrup.

2. Future meeting dates were set for

December 2 & 3	Eastern Joint Computer Conference, Boston, Mass.
February 3 & 4	A. I. E. E. New York, N.Y.

A tentative date of Oct. 14 & 15 during the A. I. E. E. meeting in Chicago was suggested. Mr. Tholstrup will forward meeting notices if a meeting will be held at that time.

3. Discussion was held on the proposal for a standard language. It was suggested that the wording of the proposal included with the minutes of the last meeting be changed as follows

2.0 Storage media used shall be 1 inch perforated tape in accordance with EIA standard RS-227.

NMAH 310/1

3.0 The eight tracks shall be numbered in accordance with EIA standards RS-227

4. After considerable discussion it was decided that a questionnaire would be sent to the Committee mailing list explaining that a standard language was being considered. That consideration was being given to the use of either the British proposal, the Flexowriter code, or the Fieldata code. That their comment regarding a proposed standard language is requested. Appendix I is this questionnaire.

5. The committee members agreed upon the following reorganization of the committee,

TR 24.4	General Storage Media and Language-Chairman Henry Tholstrup
TR 24.4.1	Punched tape Language and Format-Chairman Carl Toenebaum
TR 24.4.2	Disbanded duties transferred to P S.6
TR 24.4.3	Magnetic Tape Language and Format-Chairman Harry Hayman
TR 24.4.4	Punched Cards, Language, Format and Physical Dimensions Chairman-Henry Tholstrup
TR 24.4.5	Punched Tape, Physical Dimensions and numbering system of tracks.

6. The following is a preliminary outline of responsibilities of TR 24.4.3

A. Number of tracks for each width of magnetic tape
  (1) Spacing of tracks
  (2) Track Width

B. Recording of magnetic tape
  (1) Type of recording
  (2) Density of recording
  (3) Block length
  (4) Interblock spacing

C. Language of a single character

D. Numbering of tracks

7. Further discussion was held on the standard proposal for 11/16", 5 hole paper tape. Attempts were made to reach a compromise on the conflict resulting from the business machine practice of guiding on the three hole side versus the communications practice of guiding on the two hole side. Action on this was deferred.

8. Further discussion was held on the subject of reels for paper tape and printing on paper tape.

9. The meeting was closed at 3:00 p.m. September 15, 1959.

Respectfully submitted,

Harry Heyman

Harry Heyman

approved.

Approved

11

APPENDIX I

TR 24.4
Subcommittee on Language and Media

This subcommittee is working on a standard language, and in doing this the first step is to look at the alphabetic and numeric portions of the six bit code to determine if a fundamental basis of agreement can be reached.

In order to reduce the problem to the simplest basis only the six information bits are included. Two additional bits for parity and control will be considered at a later date. A brief listing of the advantages and disadvantages of each of the systems to be considered precedes the table.

It is requested that you indicate which system you prefer and furnish any additional information which might be pertinent in the selection of a standard language.

Fieldata Code

Advantages:

1. No discontinuities in the alphabetic portion.
2. On a binary weighted sort the numerics follow the alphabetics.
3. On a binary weighted sort the alphabetics are in alphabetic order.
4. Adoption as a military standard by the Department of Defense is likely.

Disadvantages:

1. The alphabetics are not compatible with decimal arithmetic or the Hollerith code.
2. No equipment is as yet in use, except for military experimental models.

Flexowriter/IBM-705

Advantages:

1. Zone pulses readily converted from Hollerith to binary.
2. Most commonly used, by most existing commercial equipment.
3. Adoption likely by EIA and AIA committees on numerical control.

Disadvantages:

1. In computers, cannot be sorted by binary weight.
2. Discontinuities exist in alphabetics.

British Proposed Standard

Advantages:

1. The coding is arranged in decimal sequence which permits a logical sort by binary weights.
2. Would provide an international standard if British finally adopt this proposal.

Disadvantages:

1. Not now used in United States.
2. Discontinuities exist in alphabetics.

	FIELDATA	FLEXOWRITER/IBM-705	BRITISH PROPOSAL
000001		1	1
000010		2	2
000011		3	3
000100		4	4
000101		5	5
000110	A	6	6
000111	B	7	7
001000	C	8	8
001001	D	9	9
001010	E	0	
001011	F		
001100	G		
001101	H		
001110	I		
001111	J		
010000	K		
010001	L		A
010010	M	S	B
010011	N	T	C
010100	O	U	D
010101	P	V	E
010110	Q	W	F
010111	R	X	G
011000	S	Y	H
011001	T	Z	I
011010	U		
011011	V		
011100	W		
011101	X		
011110	Y		
011111	Z		
100001		J	J
100010		K	K
100011		L	L
100100		M	M
100101		N	N
100110		O	O
100111		P	P
101000		Q	Q
101001		R	R
110000	0		0
110001	1	A	
110010	2	B	S
110011	3	C	T
110100	4	D	U
110101	5	E	V
110110	6	F	W
110111	7	G	X
111000	8	H	Y
111001	9	I	Z

CACM v2 #9 ~~Aug~~ Sept 1959

STANDARDS

EDITOR'S NOTE

While the views expressed in the paper below conflict violently with my own[1], they represent a substantial amount of constructive thinking and planning based upon solid experience in development of techniques for use in an IAL ALGOL language compiler. I feel, therefore, that they should be published for your scrutiny and that Mr. Bemer's work deserves your careful consideration.

This essay deals with single *characters* rather than with their first-order groupings into *words*, which had been intended as the primary business of this department for this year. Further, the driving force behind the Bemer alphabet has been the expressing of a particular algorithmic language, that of the Reference level of ALGOL.[2] It is intended to have quite general value, however, and has purposely included a substantial number of unassigned characters in order to permit expansion.

Succeeding issues of this Section now under preparation will include contributions to (word) glossary information by C. C. Gotlieb (University of Toronto), A. S. Householder (ORNL), W. M. Carlson (AIChE), R. P. Sell (DATAmatic), and F. J. Gruenberger (RAND), and to flowchart symbol terminology by W. B. Fritz (Westinghouse). These contributions are gratefully acknowledged. I should also like to acknowledge permission granted by D. Van Nostrand Co., Inc., to reprint portions of A. S. Householder's "A Glossary for Numerical Analysis," which is to be published as a part of the forthcoming "International Dictionary of Applied Mathematics," edited by W. F. Freiberger.

H. S. B.

A Proposal for a Generalized Card Code for 256 Characters

R. W. Bemer, *International Business Machines Corp., White Plains, N. Y.*

Introduction

There is a growing interest in expanded character sets. No matter how fast computers can be made to operate, further economy of operation can always be obtained through sound logistic usage. Input from and output to the human is taking an increasing portion of the overall time of problem solution with computers. If we had only the two digits 0 and 1 to enter into the machine and read as output, this time would be even more disproportionate. The mind, however, is capable of operating more efficiently with a larger character set of digits, 0 to 9.

The same argument applies to word symbols. Obvious inefficiencies are created when the programmer must use as GREATER THAN OR EQUAL TO rather than the single character "≧." If the human uses symbols such as these for greater efficiency in various fields such as business, mathematics, music, chemistry, etc., then the computer will be an effective working partner only when it too can utilize such symbols. Recent interest in the International Algebraic Language has only started to show the need.

We should be capable of representing larger character sets not only in internal memory and on magnetic tape, but on input/output equipment as well. The 80-column card is a logical starting point because of its volume usage as an input medium.

Design Criteria

In creating an expanded card code we must meet at least these requirements:

1. The new set must contain the existing set as a subset with exactly the same code—for character representation.
2. The new set should contain at least 256 (2^8) combinations and be expansible beyond this number, for

---

[1] Cf. Bright, H. S., A Proposed 64-Character Alphabet for Hardware Implementation of the IAL(ALGOL) Reference Language, *Comm. Assoc. Comp. Mach. 2*, No. 5, 6–9 (May 1959).

[2] ACM-GAMM Committee, Report on the Algorithmic Language ALGOL, *Numerische Math. 1* (January 1959).

eventually the maximum of 2^{12} and probably more will be needed. The present number of 256 was chosen for several reasons:

a. $256 = 2^8$, where 8 itself equals 2^3 and is thus a logical bite. This makes a group of 8 bits a likely character representation for many existing and proposed machines.

b. 8 bits for the general character may be broken into two 4-bit groups for more economical decimal arithmetic.

c. 128 is unsuitable because the ALGOL set is pushing and will shortly pass this number.

3. The code set does not necessarily need to be completely assigned to characters at this time. In fact, a sparse set has many desirable aspects.

4. The initial columnar structure should be patterned, if possible, after existing zone-numeric combinations in some manner. This may well permit minimum modification of existing hardware.

5. Codes should be constructible and reproducible on existing keypunches.

6. No code should be assigned to more than one character. No character should be represented by more than one code.

7. ALGOL characters should be included.

8. Characters not in the present IBM set or ALGOL but used by other manufacturers should be included.

9. Simple pattern of correspondence should exist between codes assigned to upper and lower case alphabetic characters.

numeric	ZONE COMBINATION													
	blank		12		11		0		12 - 11		11 - 0		0 - 12	
	blank		+	&	-	-	0	0	&	-+	$\overset{-}{0}$	0-	$\overset{+}{0}$	+0
1	1	1	A	A	J	J	/	/	a	-A	j	OJ		
2	2	2	B	B	K	K	S	S	b	-B	k	OK	s	+S
3	3	3	C	C	L	L	T	T	c	-C	l	OL	t	+T
4	4	4	D	D	M	M	U	U	d	-D	m	OM	u	+U
5	5	5	E	E	N	N	V	V	e	-E	n	ON	v	+V
6	6	6	F	F	O	O	W	W	f	-F	o	OO	w	+W
7	7	7	G	G	P	P	X	X	g	-G	p	OP	x	+X
8	8	8	H	H	Q	Q	Y	Y	h	-H	q	OQ	y	+Y
9	9	9	I	I	R	R	Z	Z	i	-I	r	OR	z	+Z
23	▨	▨							[	LB	:=	LS	;	SC
24	▨	▨					_	US						
25	▨	▨												
26	▨	▨					switch	SW	bool	BO				
27	▨	▨	{	BG							stop	SP	□	BX
28	▨	▨					‡	SY			√	SQ		
29	▨	▨							]	RB	=:	RS		
34	▨	▨			∧	LM			comm	CM			~	TD
35	▨	▨	¢	CE					≤	LE	$_{10}$	TN		
36	▨	▨			∨	LO			:	CO	←	LW		
37					(	LP			proc	PC			complex	CX
38	#	#	.	.	$	$	,	,						
39			○	CI	<	LR			cr	CR	△	TR		
45									}	ND	≠	UN		
46							↑	UW	do	DO			↓	DW
47			°	DG	±	PM			dbl pr	DP				
48	@	@	⌑	⌑	×	*	%	%	"	DQ	'	QU		
49			≡	ID					\	RD				
56					¬	NO			∀	EO				
57			≥	GE					!	EP				
58			⋕	EH					=	EQ				
59			if ei	IE					return	RE	|	VR		
67									go to	GO				
68			½	HF										
69			?	IF	or if	OR			for	FR	→	RW		
78														
79					)	RP			>	GR				
89					¼	QR					array	RY	∞	IY

representation - mnemonic multipunch or standard

character

FIG. 1. 256 character card code

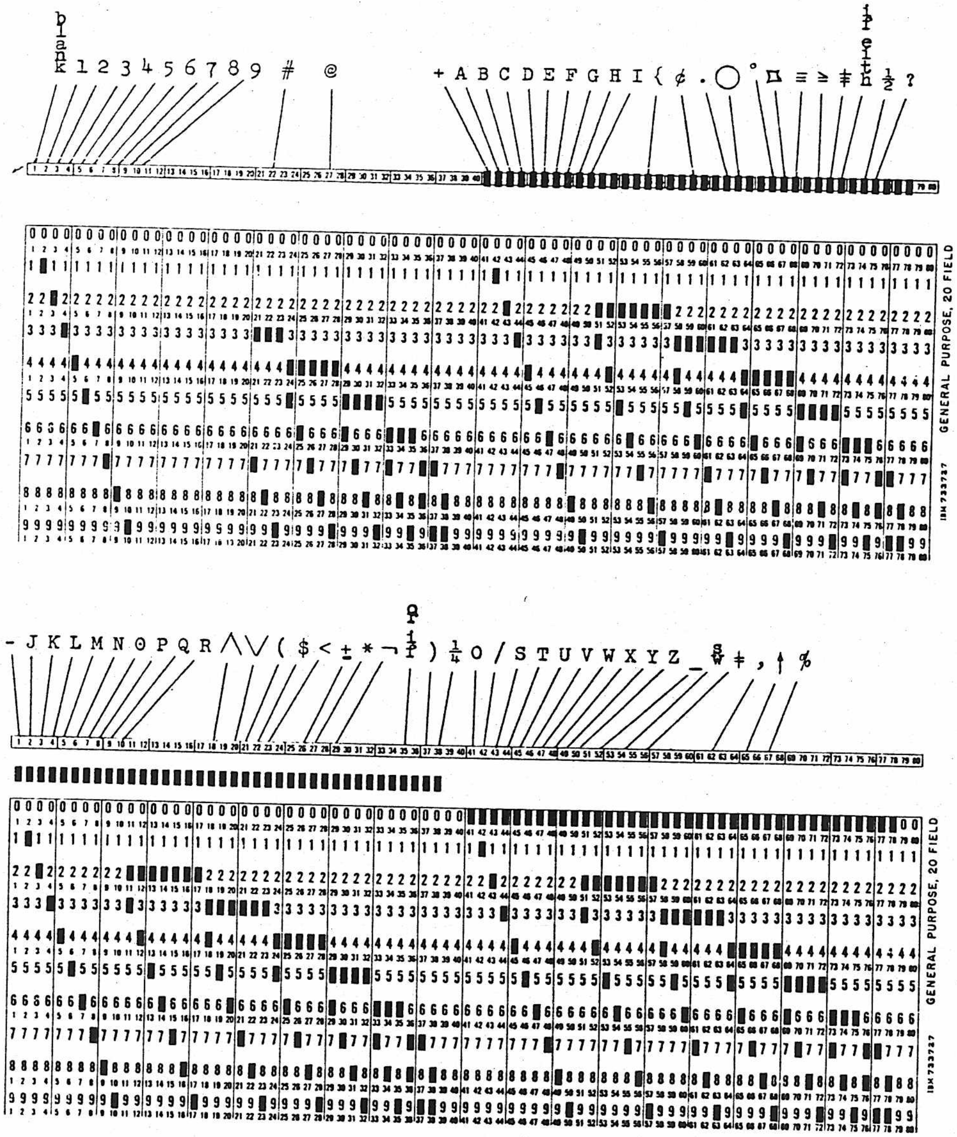
10. New characters should be assigned in logical patterns even though the full set is not used now. For example, empty code positions should be allowed in assigning the Greek alphabet, even though just a few characters are assigned now.

11. A certain number of codes should be unassigned to characters. These would be used for control operations, indefinite expansion of the set and alternate usages.

A Proposed Solution

There are 322 possible combinations of no more than four punches per column, when no more than two may be zones (12, 11, 0) and no more than two may be numeric (1 through 9). Figure 1 shows the 256 of these which remain when all combinations with two numeric punches containing a '1' punch and ten other combinations are excluded. It also shows the assignment of old and new characters to card codes. Figures 2a and 2b show the same assignment applied to actual cards.

The most important feature of this system is that each of these combinations may be represented in a single column by superimposing the codes for two alphameric characters of the present system! These two characters are chosen for their mnemonic content. Thus [is represented by LB (for Left Bracket) and is composed of the punch combinations 11-3 (for L) and 12-2 (for B), and the card code automatically assigned to represent the left bracket is the combination 12-11-2-3.



blank 1 2 3 4 5 6 7 8 9 # @ + A B C D E F G H I { ¢ . ○ ° ⊓ ≡ ≥ ≢ h ½ ?
GENERAL PURPOSE, 20 FIELD
- J K L M N Θ P Q R ∧ ∨ ($ < ± * ¬ ½) ¼ 0 / S T U V W X Y Z _ ‡ , † %
GENERAL PURPOSE, 20 FIELD

FIG. 2a

The ten paired numeric combinations were chosen arbitrarily from among those without mnemonic content to increase the odds that a new character to be added will have a unique mnemonic representation. In the few cases of conflict a new mnemonic must be chosen. Non-mnemonic codes are better reserved for control functions. Figure 3 shows the derivation of the mnemonics chosen for the new characters assigned here.

Notes and Alternatives

1. Precedence has been given to all original commercial characters except &, which gives way to +. FORTRAN characters that are currently assigned to duplicate codes are now reassigned.

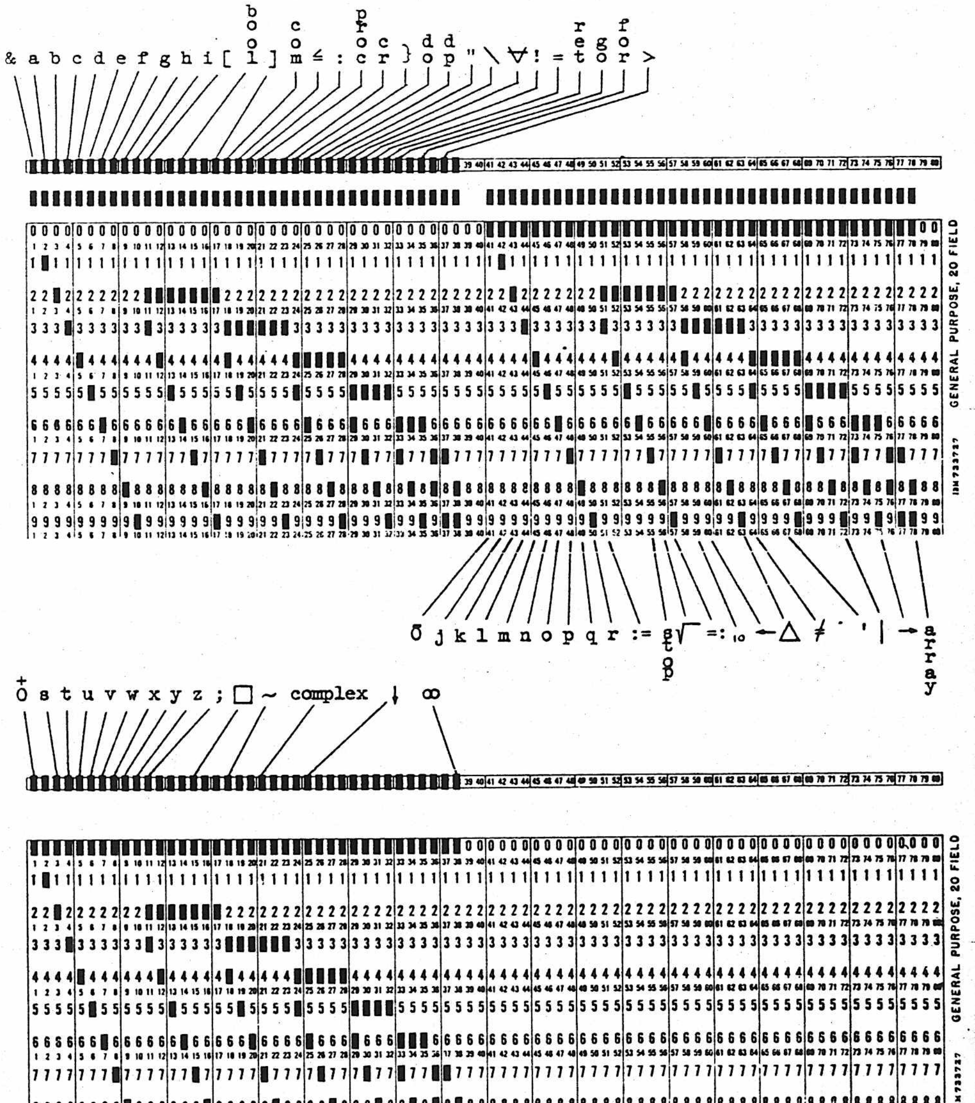
2. Certain ALGOL words are arbitrarily assigned single symbols:

if is identical to ? (* is assigned as
begin is identical to { multiply symbol
end is identical to } cause of com
integer is identical to # usage)

3. Other words could be assigned to single symbo desired:

comment could be identical to "
stop could be identical to !
return could be identical to ←

4. Record mark (⧧) and group mark (⯒) have the actual characters assigned but are not available on



b o o l
c o m
p r o c
c , d d
r e t
f o r
& a b c d e f g h i [l] m ≤ : c r } o p " \ ∀ ! = t o r >
0̄ j k l m n o p q r := g t o √ =: ₁₀ ← △ ≠ ' | → a r r a y
+
0 s t u v w x y z ; □ ~ complex ↓ ∞
GENERAL PURPOSE, 20 FIELD
IBM 723727

FIG. 2b

CHAR.	MNEMONIC	SYMBOLIZING
+	+	
-	-	
×	*	
/	/	
\	RD	Reverse Divide
±	PM	Plus or Minus
√	SQ	SQuare root
=	EQ	EQuals
≠	UN	UNequal
>	GR	GreateR
≥	GE	Greater or Equal
≤	LE	Lesser or Equal
<	LR	LesseR
~	TD	TilDe
¬	N⊙	N⊙t
∨	L⊙	Logical ⊙r
∧	LM	Logical Multiply
∀	E⊙	Exclusive ⊙r
≡	ID	IDentical to
¢	CE	CEnts
¼	QR	one QuarteR
½	HF	one HalF
cr	CR	CRedit
°	DG	DeGree
∞	IY	InfinitY
go to	G⊙	G⊙ to
do	D⊙	D⊙
return	RE	REturn
stop	SP	StoP
for	FR	FoR (F⊙R)
or if	⊙R	⊙R if
if either	IE	If Either

CHAR.	MNEMONIC	SYMBOLIZING
,	,	
.	.	
;	SC	SemiColon
:	C⊙	C⊙lon (C⊙L)
!	EP	Exclamation Point
'	QU	QUote
"	DQ	Double Quote
?	IF	IF
:=	LS	Left Substitution
=:	RS	Right Substitution
$_{10}$	TN	base TeN
(	LP	Left Parenthesis
)	RP	Right Parenthesis
[	LB	Left Bracket
]	RB	Right Bracket
↑	UW	Up arroW
↓	DW	Down arroW
←	LW	Left arroW
→	RW	Right arroW
{	BG	BeGin
}	ND	eND (END)
|	VR	Vertical
△	TR	TRiangle
□	BX	BoX
○	CI	CIrcle
—	US	UnderScore
procedure	PC	ProCedure
switch	SW	SWitch
array	RY	aRraY
comment	CM	CoMment
integer	#	
boolean	B⊙	B⊙olean
complex	CX	CompleX
double prec.	DP	Double Precision

FIG. 3. Mnemonic representation of characters

punches: the mnemonics SY and EH shown in figure 1 are for convenience in keypunching in this system and have no actual mnemonic significance. Also, the record mark might be used for "not equal" (≠).

Advantages

1. All existing 24 and 26 keypunches will reproduce correctly when these are no more than 4 punches per column, as in this system.

2. New characters can all be represented in either the single column superposition or in two adjacent columns by the mnemonic equivalent. This provides a convenient alternate for card readers which are not equipped to handle anything but the existing set. Readers which accept the column binary image can handle this system without modification.

3. The mnemonic abbreviations are a convenient interim device. When input/output equipment that will handle all of these characters finally becomes available the mnemonics will not be needed and usage will atrophy

Conclusion

An expanded character set is inevitable and must be representable in punch cards. Conversion from mechanical to electronic and other new methods will free restrictions on additions to the set. The manufacturers must therefore resign themselves to this expansion and plan for it in a standardized manner. With respect to card input equipment, the only sure way of meeting expansion is to reproduce the column image in memory as a string of bits or an address. In this address may be found the internal representation of that character. Conversion for output will be performed in the same table lookup manner. In other words, the safe way to plan input/output equipment is to avoid hardware conversions and rely entirely upon internal programming for translation.

IBM

Date: November 9, 1959
From:
Department: Applied Programming
Location: DSDHQ

To: Mr. J. C. McPherson
CC:
Department:
Location: WHQ

Subject and/or Reference: Proposed Standard Character Set

This proposal for a universal character set is the result of studies in connection with the Common Machine Language Committee and the Committee on Standards. The objective was to establish a single character set of sufficient power to be used by all future computer systems. This code was to be the external storage code as well as the internal machine code.

In arriving at the standard, people with various interests were consulted. The interrelations of input, output, storage, and communication media were studied. The size and content of the character set was investigated.

A two hundred and fifty-six character set was chosen since it seemed to be the minimum power of two, capable of expressing the characters and functions necessary for man to communicate with a machine.

The assignment of codes would seem arbitrary. However, this is not so. Special emphasis was placed upon having the natural collating sequence of letters and numbers reflected in the binary codes assigned to them. The upper and lower case alphabets are interspersed for text work. Printing subsets are obtainable without active logic by the mere stripping of one or two bits from the eight bit characters.

While this set is for future use, shortly to be announced equipment can develop and use these codes.

In order to adapt the 88 (176) character 4195 typewriter to the expanded character set, several points must be considered.

1. The character printed by the typewriter is determined by three conditions: The angular position of the ball (11 positions), the tilt of the ball (4 positions), and the shift (2 positions). In order to more closely approximate the expanded character set, a second shift value, (the red shift) is to be incorporated in the typewriter. The 88 additional characters will be indicated by printing the corresponding basic character in red.

2. An auxiliary keyboard will be used in order that the remaining 80 characters of the set will be available although on a non-print basis.

NMAH 316/1

3. The output media will be hard paper copy and standard IBM cards punched in the 8 bit representation.

4. Since the expanded character set contains subsets of 16, 48, 64, and 120 characters, the typewriter character set must map into the expanded character set directly.

5. Conversion between program typewriter and text typewriter and test typewriter configurations can be accomplished by changing the ball.

The tilt, shift, and red shift positions are binary in nature. The angular position of the ball is not. The angular position of the ball is determined by the presence or absence of four projections on the arm connected to each key. These positions were given weights of -5, 2, 2, and 1 in order to prevent overtravel of the ball. Study demonstrated that these weights can be transformed into eleven contiguous binary numbers in the range 0101 to 1111.

Further study demonstrated that the following bit assignment pattern gave a good mapping into a 256 character set:

Bit Assignment: $\underset{\text{shift}}{7}\quad \underbrace{6\quad 5\quad 4\quad 3}_{\text{ball}}\quad \underbrace{2\quad 1}_{\text{tilt}}\quad \underset{\text{red shift}}{0}$

This however gave a pattern as shown in Figure 1. If the following transformations are made in the above pattern, the codes generated by the typewriter map directly into the expanded character set as shown in Figure 2.

1. 00101xxx → 10100xxx
2. 0011xxxx → 1001xxxx
3. 0100xxxx → 1000xxxx

This appears to be a highly desirable modification and appears relatively inexpensive. For example, the three conditions can be detected by a relay network using four single-position relays and one two-position relay. Electronically, it would appear that a network containing 4 inverters, 6 two-way "and's" and 2 two-way "or's" can accomplish detection. Electronic detection and transformation before entry into the buffer is also easy to obtain. Ideally, translation should be made between typewriter and buffer. If this is not accomplished, additional circuitry will be required to get 40 characters from the keyboard in the correct position in the character set. This would necessitate either suppression of the translation circuitry if the keyboard generates true code directly or provision for transforming codes generated by the keyboard.

It would appear that a single character buffer located between the typewriter and the punch is highly desirable. Transmission of data between the buffer and punch should take place on the next key depression. This would allow correction of an error if immediately detected. If punching is not delayed in this manner, correction of any error would necessitate repunching the entire card.

This proposal is forward looking. We must of course provide linkage to the vast amounts of data and equipment using the Hollerith punched card code.

Predication of the standard code upon Hollerith card code leads to sets which violate all of the criteria upon which the standard set is founded. The mapping of Hollerith into the standard code by other than a brute force 4 x 12 x 6 matrix seems unreasonable. One-fourth of the Hollerith combinations do not lend themselves to simple conversion. Conversion to Hollerith requires a 3 x 3 x 2 matrix.

The present tape code which is a binary map of Hollerith is also unmappable. Conversion to and from it should also be handled by matrices. After working with the Hollerith and standard codes we feel that a brute force hardware method is the most desirable approach. The 705 demonstrated the impossibility of mapping Hollerith directly into a sortable binary code. This restriction is imposed by the Hollerith code in which the special characters are exceptions to the normal pattern. If the Hollerith code were decimally sortable, these problems would not arise.

The 1403 chain printer can be modified to print 48, 60, 120, or 240 character sets. These arrangements are shown in Figures 4 and 5. The 60 character subset appears to be particularly desirable. Four sets of characters can be placed on the chain for a printing speed of 480 lpm. Fewer exceptions to a natural 64 character subset must be recognized and, finally, the additional 12 characters greatly increase the utility and adaptability of the printer.

RWBemer
R. W. Bemer

H. J. Smith

F. A. Williams

Figure 1

TYPEWRITER MAPPING

	H \ L	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Auxiliary Keyboard	0000																
	0001																
	0010										$		@		%		#
Upper Case	0011		[		~		&		<		>		(		]		)
	0100										'		|		=		;
	0101																
	0110																
	0111																
Auxiliary Keyboard	1000																
	1001																
	1010										⌑		*		a		b
Lower Case	1011		c		d		e		f		g		h		i		j
	1100		k		l		m		n		o		p		q		r
	1101		s		t		u		v		w		x		y		z
	1110		0		1		2		3		4		5		6		7
	1111		8		9		-		+		.		,		/		

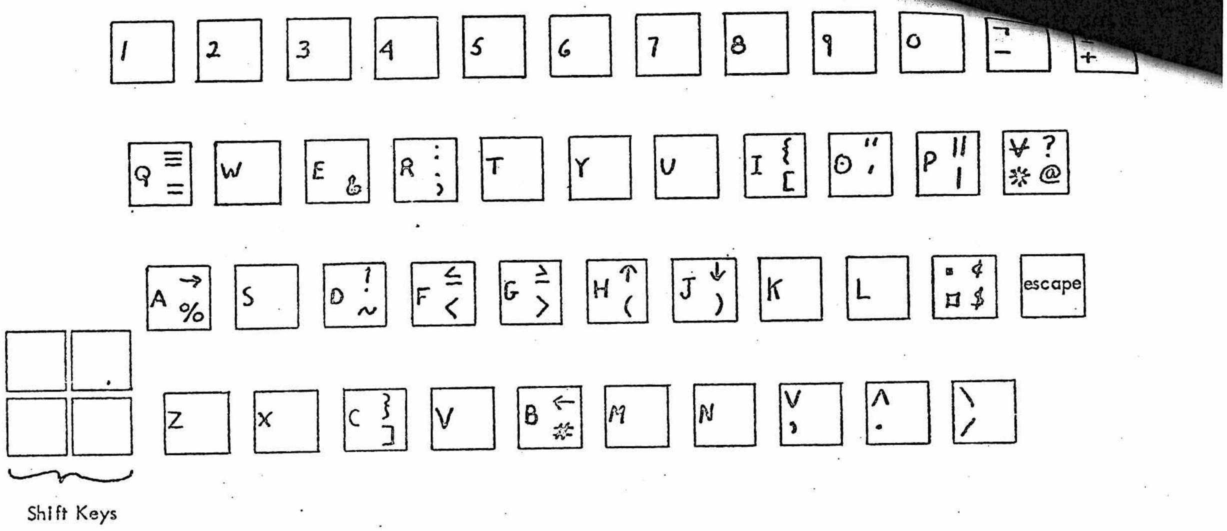
Internal, Printer, Tape I[illegible]ormation,

H \ L	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011				
0000																
0001																
0010																
0011																
0100																
0101																
0110																
0111																
1000									"	'	||	|	≡	=	:	;
1001	}	]	!	~	√	&	≤	<	≥	>	↑	(	{	⌶	↓	)
1010	¢	$	?	@	→	%	←	#								
									=		=		=		=	
	red		red		red		=		⌀	⌑	∀	✲	A	a	B	b
1011	C	q	D	d	E	e	F	f	G	g	H	h	I	i	J	j
1100	K	k	L	l	M	m	N	n	Ö	o	P	p	Q	q	R	r
1101	S	s	T	t	U	u	V	v	W	w	X	x	Y	y	Z	z
1110	0	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7
1111	8	8	9	9	¬	-	±	+	∧	.	∨	,	\	/		

Transposition Upon Typewriter Output

Figure 2

TYPEWRITER KEYBOA



Note: The above layout is tentative. The design of the keys and the location and organization of shift keys should be established by human engineering considerations.

Figure 3

										'		|		=		;
		]		~		&		<		>		(		[		)
		$		@		%		#		⌑		✳		a		b
		c		d		e		f		g		h		i		j
		k		l		m		n		o		p		q		r
		s		t		u		v		w		x		y		z
		0		1		2		3		4		5		6		7
		8		9		-		+		.		,		/		

48 character set obtained by stripping high and low order bits and suppressing codes less than 010000

60 character set obtained by stripping high and low order bits and suppressing codes less than 000100

Figure 4

PRINTER 12[illegible] CHARACTER SET

Obtained by stripping high order bit and suppressing codes less than 0001000

									"	'	||	|	≡	=	:	;
	}	]	')	~		&	≤	<	≥	>	↑	(	{	[	↓	)
	¢	$	?	@	→	%	←	#	▪	⌑	∀	✳	A	a	B	b
	C	c	D	d	E	e	F	f	G	g	H	h	I	i	J	j
	K	k	L	l	M	m	N	n	O	o	P	p	Q	q	R	r
	S	s	T	t	U	u	V	v	W	w	X	x	Y	y	Z	z
	0	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7
	8	8	9	9	¬	-	÷	+	∧	.	∨	,	\	/		

Figure 5

Minutes of Meeting
Of The TR24.4 Subcommittee
On Language and Media
Electronics Industry Association

Boston, Mass.
Eastern Joint Computer Conference

December 2 and 3, 1959

Members Present
Dec. 2, 1959

Henry Tholstrup	Commercial Controls Corp. Rochester, N. Y.
Harry Hayman	Federal Aviation Agency, Wash. D.C.
Carl Toeneboehn	Emerson Electric, St. Louis, Mo.
Gordon Burns	Teletype Corp., Chicago, Ill.
J. H. Sweer	RCA, Camden, N. J.
J. K. Nelson	Western Union, New York, N. Y.
L. L. Griffin	Navy Management Office, Wash. D. C.
Gloria Bullock	Burroughs Corp., Electrodata Div., Pasadena, Calif.
George Lyons	IBM, Poughkeepsie, N. Y.
John S. Copeland	Fairchild Graphic Equipment Plainview, L.I.N.Y.
George E. Lyons	IBM, Poughkeepsie, N. Y.
R. O. Rutherford	RCA Victor, Camden, N. J.
C. S. Roark	IBM, Poughkeepsie, N. Y.
James O'Donnell	RCA
Carlos de Villars	RCA

TR 24.1

1. The meeting was called to order at 10:00 A.M. by Chairman H. Tholstrup.

2. Mr. L. L. Griffin reported that Military Standard 188 is now in Category III, which means that this standard has been approved by the three services and is now in the final stages of adoption. Figure 19 from this Standard is attached and indicates the method of numbering holes which is a part of this Proposed Standard.

3. Considerable discussion was held regarding the proposed questionnaire on language which was included as part of the minutes of the meeting of September 14 and 15, 1959. It was finally agreed that interested parties would be given an opportunity of commenting on various languages

6756:1-5:1/60(TR-24.4)

NMAH 310/1

and proposals at the next meeting to be held in New York City during the week of February 1 - 5.

4. The latest Fieldata Code is attached with its Baudot equivalents. Also attached is a letter from E. F. Carlberg, Chairman of the Numerical Control Panel of the Aircraft Industry Association indicating their soon-to-be-published National Aircraft Standard for Numerical Control.

TR 24.4.3

1. The following statement of purpose was agreed upon for TR-24.4.3.

> "To establish standards for digital magnetic tapes to facilitate intercommunications between machines of various manufacturers."

2. Mr. Roark then described some of the various methods of recording on magnetic tape.

3. Appendix IV indicates the preliminary draft of a questionnaire to be distributed by this committee to assemble the necessary information on magnetic tape.

TR 24.4.1

1. The following is the status of the proposal on hole numbering as of November 24, 1959. The proposal was essentially identical to Figure 19 of Mil Standard 188.

Approvals	30
Approvals with comments	3
Disapprovals	10

Companies disapproving:

Jones and Lamson Machine Co.
Commercial Controls Corp.
Bendix Computer Division
IBM, Poughkeepsie
Friden, Inc.
Emerson Electric Mfg.
Remington Rand, Norwalk, Conn.
Boeing Airplane
Cincinnati Machine Milling
General Electric Co., Waynesboro, Va.

The entire file on hole numbering will be forwarded to TR 24 for action and recommendation.

2. The discussion on guiding edge for 11/16" paper tape was continued with IBM representing the discussion of guiding on the 3 hole side and Teletype discussing the guiding on the 2 hole side. It was finally agreed that the proposal would be forwarded for vote with comments by Mr. George Lyons and Mr. Gordon Burns. Three options would be provided for the guiding edge for a vote.

Option A	2 level side	.300 maximum
	3 level side	.395 maximum
Option B	.392 ± .003 (three hole side)	
Option C	.297 ± .003 (two hole side)	

3. The meeting was adjourned at 2:45 P.M., December 3rd, 1959.

Respectfully submitted,

Harry Hayman

Harry Hayman

Teleprinter for Reliable Transmission of Numbers*

By J. AUGUSTIN

Standard Elektrik Lorenz AG; Stuttgart, Germany

PRIVATE and public administrations often transmit numerical teleprinter information. When straight text is transmitted, error consisting of a wrong letter is most often quickly apparent and the correct letter can be deduced from [illegible] context. It is quite different with numbers. The receiving subscriber usually has no means of perceiving erroneous numbers replacing them by the correct numbers. Errors can be caught by complete retransmission of each individual message; visual inspection of the two versions of the message will reveal any errors. However, this method is rather costly, especially where much information is handled.

There was [illegible] reason, then, to develop a teleprinter for reliable transmission of numbers. Compatibility of such a machine with the public telex systems is of particular importance; in the majority of cases, maintenance of a private network is too expensive.

The machine developed has three shift positions instead of the conventional two. The first two shifts comprise the usual lower- and upper-case characters of any ordinary teleprinter. The third shift provides reliable transmission of 15 characters: The numbers through 9, plus sign, minus sign, space, carriage return, and line feed. Of the 32 possible combinations in the 5-unit code, only those 15 were selected that are characterized by 1 and by 3 marks in the 5 positions. Hence, at least 2 elements of a combination have to be wrong to cause an error. In principle, combinations having 2 and 4 marks in the 5 positions could also have been selected. However, the combinations for carriage return, space, and line feed also must be used in the first and second shifts. In telegraph systems operated according to Comité Consultatif International Télégraphique et Téléphonique recommendations, these three combinations are of the group having 1 mark in the 5 positions; this decided the case in favor of the 1-mark and 3-mark combinations.

Figure 1 shows a teleprinter providing reliable

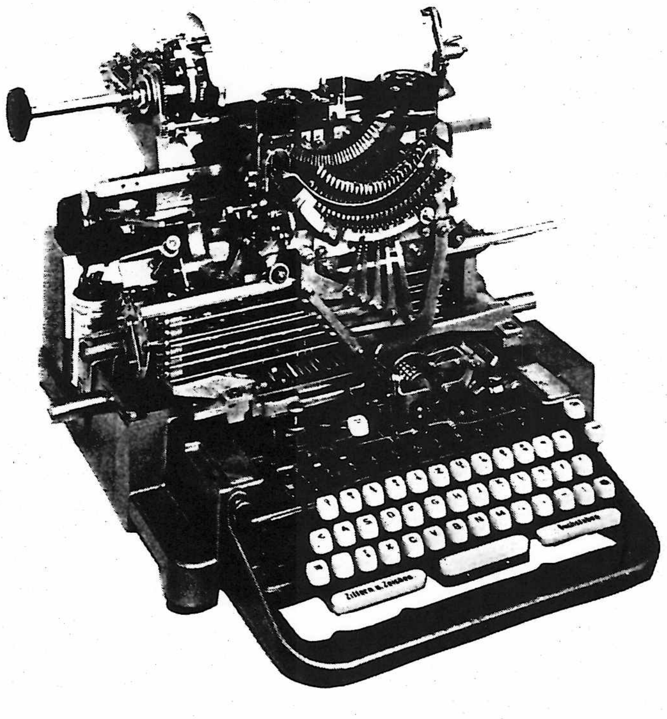


Figure 1—Teleprinter for reliable transmission of numbers. The set of black keys used in the third-shift position is visible above the conventional keyboard.

---

*Originally published under title "Fernschreiber zur gesicherten Übertragung von Ziffern" in *SEG-Nachrichten*, volume 6, number 1, pages 58–[illegible] 1958.

vol 35, #4, 1959 (no month)

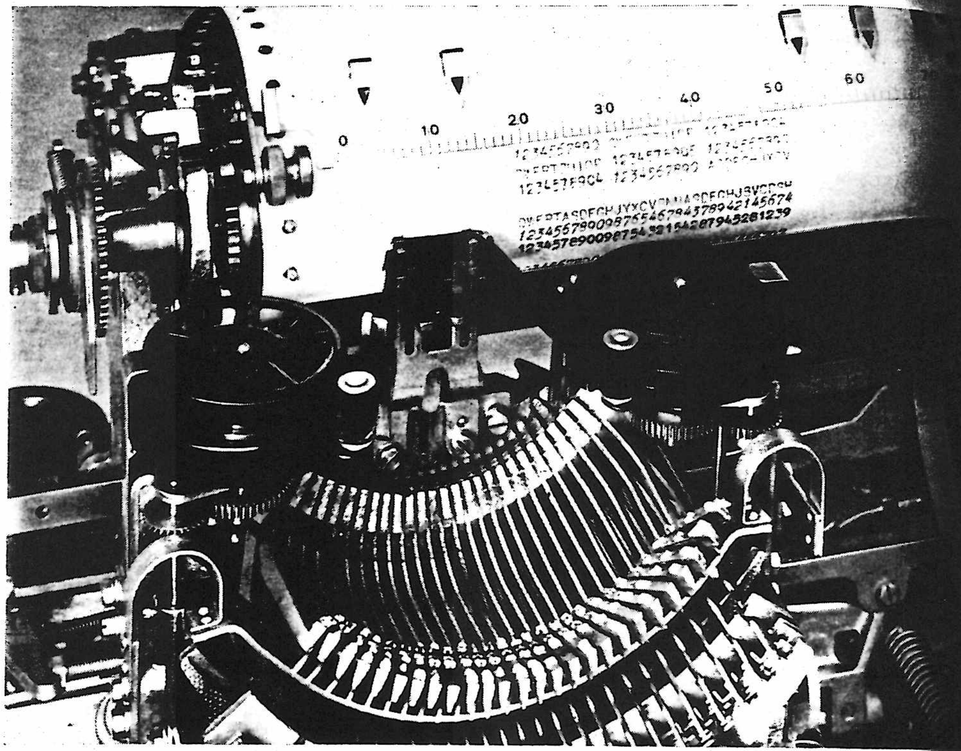


Figure 2—View of the type bars with three characters each and the teleprinted message in which denote the high-reliability numbers.

transmission of numbers. The conventional keyboard has white keys. Above it are another 12 back keys for 0 to 9, plus, and minus. A locking bar releases the white keys only in the first- and second-shift positions; the black keys only in the third-shift position. Accidental operation of the wrong keyboard is thus precluded.

Each of the type bars carries three characters. (Figure 2). The position of the platen in any of the three shift positions determines which of the three characters of a type bar will print.

The numbers transmitted in the third shift are clearly distinguished from the conventional numbers by printing in italics. All type bars not equipped with the third-shift characters have an asterisk in the third-shift position. This asterisk signals that an erroneous number-combination has been received; as a result, no special electrical check measures are necessary.

If its only current impulse is dropped, each of the 1-mark-in-5-position combinations becomes combination *32* of International Telegraph Alphabet *2*; that is, the combination third shift. Since the receiver coul mine whether this code is transmitte or is just a faulty combination, a lo has been attached to release the thi only when the carriage position is at ning of a new line. In addition, a spec has been provided to print an asteris the combination for the third shif (except at the start-of-line position faulty character converted into com can be identified.

The above arrangements offer th tection against mistakes as messag in a conventional teleprinter circui they save the considerable cost tha incurred by the doubled transmissio

As mentioned before, this teleprin in the conventional manner when in the second shift and can be emplo municate with every subscriber in a network.

ART-ST
tion may
It is desig
ion by
p signals
ining the
It is t
transmissi
red mess
field of a
mission
re are two
lost by di
concerns
ation in w
teleprinter
ulses caus
that par
subseque
itted.
the receivin
re stop pu
rough an
l characte
use of th
of which
after an
tly. As a
cters prin
erably re
other ca
ich the c
lost in tr
ing set
itting en
in plain-
the en
ring at t
ter basis
throws
deciphe
re, bec
ing trans
ed.

ROBERT STEENECK, Assistant to Manager—Data Equipment and Systems

Error Checking Possibilities Concealed Within the 5-Unit Code

ALTHOUGH the 5-unit Baudot code has withstood the test of time in the printing of telegraphic messages, it appears susceptible to some improvement if this same code is to be used for data communication. Analysis confirms that when the code was established the simpler combinations were assigned to the most used characters and printer functions. If the combinations are arranged in order with the simplest combinations first, as shown in Figure 1, it will be noticed that the assignment follows quite closely the ETAOIN SHRDLU CMFWYP layout used on the linotype keyboard which was arranged in the order of frequency of language use of alphabet letters. Although this assignment has been quite satisfactory for telegraphic plain text messages considerable improvement can still be made, as will be pointed out later on in this article.

The assignment of code combinations to the ten numerical digits, however, quite definitely is not well suited for data handling. It obviously was determined by the combination already found on the top letters line of the standard typewriter keyboard, namely QWERTYUIOP. In telegraphic communication, the disadvantages of this assignment were never very objectionable because, first of all, figures were generally spelled out as words in message texts. If figures were included in a telegram confirming figures always were sent at the end of the message for an accuracy check.

In data transmission, however, figures are usually the most important part of the information being transmitted, because figures generally represent quantities of something which in the end is the equivalent of quantities of money. An undetected figures error in data transmission is therefore the most serious error that might occur within the message structure.

A quick analysis of the figures code assignments will show why this present code arrangement is not entirely satisfactory for data purposes. Let us examine the digit 1, for instance. Digit 1 has pulses 1, 2, 3 and 5 marking and pulse number 4 spacing. Now if we subject this combination to the most frequent type of error,

	3				5		8		9			4				7				2		0			6					1	
BLANK	E	LF	SP	CR	T	A	I	N	O	Z	S	R	H	D	L	U	C	M	B	W	J	P	F	G	Y	K	V	X	FIG	Q	LTRS
	1					1				1	1			1		1			1	1	1		1		1	1		1	1	1	1
		2				2	2					2			2	2	2			2	2	2		2		2	2		2	2	2
			3				3	3			3		3			3	3	3				3	3		3	3	3	3		3	3
				4				4	4			4		4			4	4	4		4		4	4		4	4	4	4		4
					5				5	5			5		5			5	5	5		5		5	5		5	5	5	5	5

5 Singles

10 Doubles

10 Triples

5 Quads

Figure 1.

vol 14 #2
western Union Technical Review

namely, the loss of a single marking pulse, we of course will get an error. If the error we get produces an asterisk, a question mark, or some other upper case character, the error obviously can be detected easily. In all of the four possible errors produced by the loss of a single marking pulse the digit 1, however, is always converted into *another digit.* Thus digit 1 by the most common of errors can become either 2, 6, 7 or zero, depending upon which pulse was lost. In fact, under actual tests, when handling figures alone, the standard Baudot code produced on the average one such undetected error for every six errors that were obvious.

This shortcoming of the Baudot code can be remedied by reassigning some 6 or 7 of the 10 digits to new code combinations. Referring to Figure 1, it will be noticed that the 32 code combinations can be divided into 5 groups according to the number of marking pulses contained in each combination. It is quite obvious that, in any one group, it is impossible to create any error that will produce any other character *in the same group,* unless there is both gain and loss of the same number of marking pulses in the particular errored combination.

This, of course, is the heart of the fixed ratio code error checking technique used in both the IBM transceivers and the RCA overseas radio error detection system. It will be noticed from Figure 1 that there are two fixed ratio groups each containing ten combinations in the 5-unit Baudot code. If the ten digits are assigned completely to either of these groups, loss of marking pulses alone can never print wrong *digits,* but instead can print only meaningless upper-case combinations that may easily be recognized as errors.

The loss of a complete digit, of course, is not prevented by any code arrangement. This problem, however, becomes a simple matter of message format quite common in many data systems. Zeros or other marks are often used to fill out numerical data groups so that all will contain predetermined numbers of numerical characters. The loss of a complete digit then becomes quite obvious. As in all error detecting schemes, it is also quite possible to fool this system too, but either pulses must be received in the wrong slots or pulses must be both lost and gained to produce an error detection failure.

Some Test Results

In the laboratory, tests were run to determine how often such a breakdown of check would occur if, first of all, pulses were received in the wrong slots. In this test the receiving printer was set on the very edge of its range on a marginal line. Even with this unusual condition over 100 errors were detected before one nondetected errored combination was produced. In the second case, a marginal circuit was used subject to both dropout and gains of pulses. Even under this condition 500 errors were recorded before one error that would not be detected was produced. These tests obviously are not conclusive but indicate the merit of the system even under adverse conditions. Under normal conditions the system has already proved its merit in the IBM transceiver and the RCA overseas radio error detecting system.

Special reassignments of 5-unit digit combinations to minimize errors have been used occasionally by both Western Union and the International Business Machines Corporation. In Europe, however, teleprinters for reliable transmission of numbers have been developed by both the Lorenz and Siemens companies. The Lorenz printer designed for Telex operation is unique in that the protected numbers are printed in italics from a third shift level built into the printer. In this level all combinations outside the three-marking two-spacing group, assigned to protected figures, print an asterisk easily recognized as a received error. The other two shift positions have standard teleprinter code assignments and can operate with machines not equipped with the third shift feature.

The Lorenz keyboard has a fourth row of keys assigned only to the protected numbers. This row is locked out until the third shift key has been depressed.

In the Siemens' system no compatibility with standard code operation has been provided. All but eleven of the 32 5-unit code combinations have been completely reassigned to obtain the desired protection of numerical information. As in the Lorenz system the three-marking two-spacing code group has been used for this purpose.

The European committee working on telephone and telegraph standards (C.C.I.T.T.) is now considering proposals to make some code change, employing fixed ratio combinations for numbers, standard for 5-unit data handling.

In alphabetic code combinations, the order of assignment is not too important since plain text communication errors generally result in unintelligible words.

Some improvement in character assignment could be made by bringing all vowels into one of the groups shown in Figure 1. An errored vowel would then most generally result in the printing of a consonant which would make the word unpronounceable and would be easily recognized as an error. Thus "six brown cups" could not become "six brown caps."

An analysis of consonants also might show the desirability of grouping those with somewhat similar sounds so that errored consonants would again tend to produce nonsensical words. To do this of course would involve a detailed study of word structure.

A Practicable Program

If in the Baudot code we reassign some six or seven digit combinations only we can obtain simple effective error protection for all numeric information, which is an essential requirement in data systems. To make such a conversion on all teleprinters would be a great task but even this would be nowhere near the task involved in converting to 7- or 8-unit fixed ratio data codes.

There is, however, no need at present to convert the entire telegraph world. Data transmission is now mostly over private wire systems where relatively few pieces of equipment are involved. The Type 28 page teleprinter can be converted quickly to accept almost any 5-unit code by merely inserting a new box of type in the printer. In keyboard transmissions, four-bank keyboards have already been designed with lockout features that will allow either the proposed data code transmission or present telegraph code transmission to be generated from the same keyboard.

Should it become necessary to handle data over the Western Union public message system, only the preamble and the end of the message must be in the present Baudot code. The body of the transmission may be any jumble of 5-unit combinations provided these combinations do not produce the cutoff signal which at present is two consecutive carriage-return combinations. In some private wire systems, Figs-H-Ltrs or a combination of four consecutive N's are used to signal the end of a message. It is possible to avoid all of these conflicts by using the three-marking pulse combination group for the ten digits. This involves reassignment of only six digits to produce a simple effective numerical error detecting code. The digits are 1, 3, 4, 5, 8 and 9.

It should be realized, however, that this change only scratches the surface in tapping the error detection potentials that now lie hidden in the 5-unit permutation code.

References

1. D-B ALPHABET NR.1, 9Fs, Ubs 1501/1 BL.3. Siemens Wernerverk fur Telegrafen und Signaltechnik.

2. TELEPRINTER FOR RELIABLE TRANSMISSION OF NUMBERS, S. AUGUSTIN, *Electrical Communications*, Vol. 35, No. 4, 1959, pp. 245-246.

FIG. 13

	1 2 8	1 8	1 2	1	8	2	BINARY CODE		↓
	3 2 1 H	3 2 H	3 1 H	3H (820)	2H (822)	1H (825)	PRINT MAGNETS		
7			G	E	⌑	F	D	4-2-1R	Z_1Z_24
6	•	I	C	A	H	B	Ɛ	4-2R	$Z_1 Z_2$
5			P	N	✳	0	M	4-1R	$Z_2 4$
4	$	R	L	J	Q	K	—	4R (785)	Z
3			X	V	%	W	U	2-1R	$Z_1 4$
2	,	Z	T	/	Y	S	0	2R (788)	Z_1
1			7	5	@	6	4	1R (791)	4
HOME POSITION 0	#	9	3	1	8	2			
	6	5	4	3	2	1	0 HOME POSITION	(BANDS)	

FIG. 14

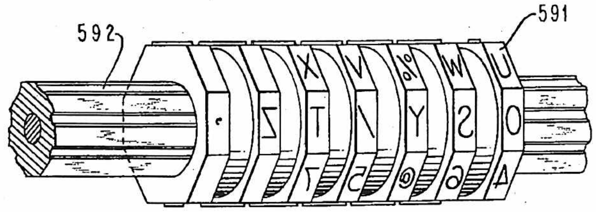


FIG. 15

PRINT MAGNETS	TOTAL TYPE	MOTION CYLINDER
Z_1 $Z_2$1	3H	6R
Z_1 $Z_2$2	1H	6R
Z_1 $Z_2$1 2	4H	6R
Z_1 $Z_2$4	0H	7R
Z_1 $Z_2$1 4	3H	7R
Z_1 $Z_2$2 4	1H	7R
Z_1 $Z_2$1 2 4	4H	7R
Z_1 $Z_2$8	2H	6R
Z_1 $Z_2$1 8	5H	6R
$Z_2$1	3H	4R
$Z_2$2	1H	4R
$Z_2$1 2	4H	4R
$Z_2$4	0H	5R
$Z_2$1 4	3H	5R
$Z_2$2 4	1H	5R
$Z_2$1 2 4	4H	5R
$Z_2$8	2H	4R
$Z_2$1 8	5H	4R
Z_1 2	1H	2R
Z_1 1 2	4H	2R
Z_1 4	0H	3R
Z_1 1 4	3H	3R
Z_1 2 4	1H	3R
Z_1 1 2 4	4H	3R
Z_1 8	2H	2R
Z_1 1 8	5H	2R
1	3H	0R
2	1H	0R
1 2	4H	0R
4	0H	1R
1 4	3H	1R
2 4	1H	1R
1 2 4	4H	1R
8	2H	0R
1 8	5H	0R
Z_1	0H	2R
Z_1 $Z_2$1 2 8	6H	6R
$Z_2$1 2 8	6H	4R
Z_1 1 2 8	6H	2R
1 2 8	6H	0R
Z_1 $Z_2$4 8	2H	7R
$Z_2$4 8	2H	5R
Z_1 4 8	2H	3R
4 8	2H	1R
Z_2	0H	4R
Z_1 1	3H	2R
Z_1 Z_2	0H	6R
	0H	0R

6/UB 1/52

B.

R+D File: ~~Language~~ Code Standards for Punched Tape

NMAH 310/1/1

Minutes of the Meeting
# of the TR 24.4 Subcommittee
On Language and Media
Electronics Industry Association

E.I.A. Headquarters
New York, N. Y.

Feb 1 and 2, 1960

Members Present

Harry Hayman	Federal Aviation Agency, BUR R & D
Robert W. Green	Standard Register Co., Dayton, Ohio
Richard H. Hunter	Electrical Systems Engineering National Cash Register Co., Dayton, O.
John L. Pfaff	Burroughs Corp., Paoli, Pa.
Geo. E. Lyons	IBM - Poughkeepsie, N. Y.
John K. Nelson	Western Union, New York, N. Y.
George M. Ives	8433 Fallbrook Ave., Canoga Park, Calif. Ramo Wooldridge, L. A.
J. H. Sweer	RCA
Wallace Velander	Navy Management Office, Wash. 25, D. C.
L. L. Griffin	" " "
Joseph D. Koenig	CDC- Minneapolis, Minn.
Henry S. Forrest	CDC - Washington, D. C.
Robert S. Boykin	U. S. Army Signal Corps Eng. Lab., Fort Monmouth, N. J.
William F. Luebobert	" " " " "
Richard M. Gottlieb	" " " " "
Hugh G. McEvoy	Federal Aviation Agency - Region One
Raymond C. Smith	Minnesota Mining & Mfg. Co. 400 McKnight Road, St. Paul, Minn. Dept.
Robert Baurley	General Electric, Missile & Space Vehicle/ 3198 Chestnut St., Phila. 4, Pa.
Henry Tholstrup	Commercial Controls
A. Craig Reynolds, Jr.	Stromberg Division of General Time Corp. Thomaston, Conn.
E. H. Bosman	IBM, Endicott, N. Y.
G. Warren Patterson	Computer Div., Philco Corp. 3900 Welsh Rd., Willow Grove, Pa.
Charles D. Whitaker	Electronic Engineering Div., NCR Dayton 9, O.
Ed Lieblein	USASRDL, Fort Monmouth, N. J.
R. W. Bemer	IBM
C. Toeneboehn	Emerson Electric, St. Louis, Mo. 19th & Alleghany - Philadelphia, Pa.
Eric H. Clamons	Remington Rand Univac
Feb. 2, 1960	
G. K. Burns	Teletype Corp., Chicago
W. F. Luebbert	U. S. Army Signal R&D Lab.,Ft. Monmouth
T. J. McNamara	Minn. Honeywell- Datamatic, Newton, Mass.
Paul P. Boehm	RCA 13-1 Camden, N. J.

1. Meeting was called to order at 9:00 A.M. by Mr. H. Tholstrup, Chairman.

2. The presentations were given in the following order and are attached as appendices to this report:

   A. Warren Patterson, Product Planning, Philco

   B. George M. Ives, Ramo-Wooldridge

   C. Joseph Koenig, Control Data Corporation

   D. William F. Luebbert, U. S. Army Signal Research & Development Laboratory

   E. R. W. Bemer, I.B.M., White Plains, N. Y.

   F. Paul Boehn, R.C.A.

   G. Thomas McNamara, Minneapolis Honeywell, Datamatic Division

   H. Dick Hunter, National Cash Register

   I. Charles Whitaker, National Cash Register

3. The next meeting will be held in New York on March 23 and 24, 1960. The following meeting will be held in San Francisco during the week of May 2 to 5.

4. Mr. Gordon Burns moved that this committee recommend to TR-24 that no subcommittee within TR-24 adopt a standard language proposal until the study by TR24.4 on language has been completed. The motion was seconded and unanimously approved.

5. The questionnaire and proposed standard for 11/16 inch paper tape will be prepared by Mr. Gordon Burns and forwarded to Mr. Hayman for legal coordination and distribution by E.I.A. headquarters.

Respectfully submitted,

Harry Hayman

Harry Hayman

A COMMON LANGUAGE CODE FOR COMMUNICATIONS AND INFORMATION PROCESSING SYSTEMS

William F. Luebbert
U.S. Army Signal Research and Development Laboratory

A common code language is urgently needed in the communications and data processing industries as a medium of information exchange. The operational and economic advantages of being able to exchange information readily when required are unquestionable; and there is general agreement that the adoption of a common language would benefit everyone, even those who for one reason or another decide not to use it. However, until recently there has been little agreement upon what code or codes should be adopted as a common language, under what circumstances the use of the common language should be manditory, or upon the concepts for transition from the present babel of data languages.

This article describes the development and design of a coding system developed by the Communications Department, U.S. Army Signal Research and Development Laboratory for use as a common language for data processing and data communications systems including advanced design high speed teletypewriter systems. The general pattern and alphanumeric code have received widespread acceptance and has been undergoing standardization action by the joint services committee on communications systems technical standards. It has been adopted by a number of large scale communications and data processing systems such as the U.S. Army FIELDATA system and the U.S. Air Force SACCS system (Project 465L). Among the manufacturers now engaged in developing data processing or communications equipment for use in systems using this code as a common language are Ampex, Bendix, Burroughs, Collins, Ford/Aeronutrc General Telephone/Sylvania, I.B.M., I.T.T., Philco, and Smith-Corona.

NMAH 310/1
(EIA Feb 1966)

The effort which led to the design of this coding system started as a fundamental examination of the effects and influence of the data code(s) used upon the characteristics and limitations of typical communications and data processing systems. The intent was to provide a rational basis to determine whether the teletypewriter signalling code (hereafter referred to as the Baudot code) or some other code in widespread use was suitable for use as a common language code for advanced data processing/communications systems such as UNICOM, CRITICOM, FIELDATA etc. This examination showed that many of the serious limitations of existing and planned large-scale communications and data processing systems were due to information handling difficulties related to the coding system used. The factors of code design found to be of particular importance in determining system constraints and lim itations were as follows:

1. RANGE AND FREEDOM OF EXPRESSION
   To minimize system constraints the code used must have a flexible and adequate "vocabulary" of code symbols with adequate provision for system supervisory and control information as well as for the information being communicated or processed. Provisions for planned extension of the code to include specialized sub-languages or "technical jargons" to meet specialized requirements not efficiently satisfied by the basic language is also important.

2. SIMPLICITY AND EASE OF MANIPULATION
   System and equipment complexity is strongly influenced by the ease of manipulation of the code. A code which is logically organized to permit different types of information (e.g. letters and numbers) to be readily segregated by simple tests rather than by complete decoding is advantageous. Various other features which simplify basic data handling operations such as sorting, alphabetizing, indexing etc. may also be valuable.

3. FLEXIBILITY OF FORM
   Various types of communications channels, recording devices and processing equipment have differing characteristics which create legitimate requirements for differences in data format, control and supervisory features and error control. An important characteristic of a common language is the ability to adapt and conform to these various requirements without loss of its common language features.

4. SPECIAL COMPATIBILITY TRANSLATION AND DECODING FEATURES TO CONFORM WITH EXISTING STANDARDS
   Any common languages code must operate with a variety of existing codes such as the various data processing codes (I.B.M., UNIVAC, RCA 501 etc.) and the various data communications codes (Baudot, I.B.M. 4-of-8, RCA R/Q 4 of 7 etc.) Features which increase compatibility with these codes or simplify and facilitate translation to and from them may be very important. Certainly the widespread use of existing 60, 66, 75, and 100 wpm teletypewriter equipment must be considered. Other standards such as the standard typewriter keyboard may also be important because of the effects of interaction of these standards and the code upon equipment design and complexity.

SCREENING CRITERIA
USED FOR DETERMINING SUITABILITY OF EXISTING CODES FOR COMMON LANGUAGE USE

Because of the obvious advantages associated with the adoption of a code already in widespread use a set of screening criteria were established to evaluate the suitability of various existing codes for common language use. These criteria were as follows:

CRITERIA ASSOCIATED WITH RANGE AND FREEDOM OF EXPRESSION

1. The common language coding system shall provide not only a common alphabetic/numeric symbols for the exchange of information among users of the system, but also a basic group of system supervisory and control symbols.

2. The alphabetic/numeric portion of the coding system shall provide a basic range of expression equal to or greater than the most widely used data codes. The IBM card (Holerith) code and the teletypewriter signalling (Baudot) code will be used as standards of comparison. Since this basic range will not always be adequate to meet specialized requirements (e.g. a need may exist for special mathematical symbols, weather symbols etc.), provision should be made for expansion of this basic range of expression. It is desirable, but not manditory, to make provision for optional capital/lower-case capabilities. It is also highly desirable to have a means of permitting the temporary or permanent creation of specialized subsidiary alphabets or "technical jargons" which can be called into usage in a predesignated way.

3. The control and supervisory portion of the coding system shall provide a basic range of expression adequate for normal control and supervision of a communications or data processing system when the user information is incomprehensible. (e.g. in arbitrary bit stream form, enciphered, or in some code other than the common language code). Provisions for expansion of the range of expression of this information similar to the provisions for expansion for user information (see paragraph 2 above) are also desirable.

CRITERIA ASSOCIATED WITH SIMPLICITY AND EASE OF DATA MANIPULATION

4. In order to preserve equipment simplicity all basic code symbols should contain the same number of bits. This requirement shall not however prohibit the use of compound symbols which contain an integral multiple of the number of bits used for basic symbols.

5. For simplicity of handling and manipulation each of the basic code symbols should have a unique bit combination.

6. In order to simplify both equipment and information processing procedures the common language code should have a logical pattern of symbol assignments which permits the easy segregation of symbols into classes without the necessity of completely decoding each symbol. For example, it should be possible to segregate system control and supervisory symbols from user alphabetic/numeric symbols by testing one (or at most two) bit(s). Similarly it should be possible to segregate letters from numbers or capital letters from lower case letters by equally simple tests.

- 3 - (Appendix D)
SCREENING CRITERIA
USED FOR DETERMINING SUITABILITY OF EXISTING CODES FOR COMMON LANGUAGE USE

Because of the obvious advantages associated with the adoption of a code already in widespread use a set of screening criteria were established to evaluate the suitability of various existing codes for common language use. These criteria were as follows:

CRITERIA ASSOCIATED WITH RANGE AND FREEDOM OF EXPRESSION

1. The common language coding system shall provide not only a common alphabetic/numeric symbols for the exchange of information among users of the system, but also a basic group of system supervisory and control symbols.

2. The alphabetic/numeric portion of the coding system shall provide a basic range of expression equal to or greater than the most widely used data codes. The IBM card (Holerith) code and the teletypewriter signalling (Baudot) code will be used as standards of comparison. Since this basic range will not always be adequate to meet specialized requirements (e.g. a need may exist for special mathematical symbols, weather symbols etc.), provision should be made for expansion of this basic range of expression. It is desirable, but not manditory, to make provision for optional capital/lower-case capabilities. It is also highly desirable to have a means of permitting the temporary or permanent creation of specialized subsidiary alphabets or "technical jargons" which can be called into usage in a predesignated way.

3. The control and supervisory portion of the coding system shall provide a basic range of expression adequate for normal control and supervision of a communications or data processing system when the user information is incomprehensible. (e.g. in arbitrary bit stream form, enciphered, or in some code other than the common language code). Provisions for expansion of the range of expression of this information similar to the provisions for expansion for user information (see paragraph 2 above) are also desirable.

CRITERIA ASSOCIATED WITH SIMPLICITY AND EASE OF DATA MANIPULATION

4. In order to preserve equipment simplicity all basic code symbols should contain the same number of bits. This requirement shall not however prohibit the use of compound symbols which contain an integral multiple of the number of bits used for basic symbols.

5. For simplicity of handling and manipulation each of the basic code symbols should have a unique bit combination.

6. In order to simplify both equipment and information processing procedures the common language code should have a logical pattern of symbol assignments which permits the easy segregation of symbols into classes without the necessity of completely decoding each symbol. For example, it should be possible to segregate system control and supervisory symbols from user alphabetic/numeric symbols by testing one (or at most two) bit(s). Similarly it should be possible to segregate letters from numbers or capital letters from lower case letters by equally simple tests.

6984:13-25

7. In order to simplify important data handling operations such as alphabetizing, sorting, merging, collating, indexing, information retrieval etc. the code combinations for letters of the alphabet should appear in consecutive order with no gaps or insertions of other symbols between the letters.

8. In order to simplify mathematical operations and numerical data handling operations such as sorting the code combinations for decimal digits should appear in consecutive order beginning with the symbol for zero with no gaps or insertions of other symbols between the digits. It is desirable that four predetermined bits give the binary coded decimal value of each numerical symbol. Excess-three binary coded decimal would be acceptable in lieu of normal binary coded decimal.

CRITERIA ASSOCIATED WITH FLEXIBILITY OF FORM

9. The common language coding system should provide for gradual degredation of range of expression as more abbreviated forms of the code are used. An example of gradual degredation would be an eight bit standard form which could be reduced to a seven bit form by dropping a parity bit used for local error control. This seven bit form might contain both user alphabetic/numeric symbols and system control and supervisory symbols. Reducing one more bit to a six bit form one might have only an alphanumeric alphabet complete with a full complement of punctuation symbols or perhaps an alphabet plus a complete selection of control and supervisory symbols. Reducing once again to five bit form one might retain just the alphabet or just basic system control and supervisory symbols. This property of gradual degredation through modular structure provides flexibility which enhances the value of the common language code by permitting incorporation of parts of it in the code languages of systems which would not otherwise find it feasible to use the common language.

10. Modularity of redundancy to permit the data form to conform to the particular error control requirements of a given communications channel, recording medium or processing device is also highly desirable. Any coding system which integrates redundancy into the basic structure of the code so tightly that decoding is necessary to remove the redundancy should be avoided.

SPECIAL FEATURES ARISING FROM REQUIREMENTS FOR COMPATIBILITY WITH EXISTING SYSTEMS AND STANDARDS

11. Because of the widespread use of the Baudot teletypewriter signalling code in existing communications systems the maximum degree of compatibility with this code consistent with meeting other criteris is required. As an absolute minimum of compatibility translation from Baudot to the common language code and back again should be a completely reversible one-for-one mapping process which retains character count integrity through the translation retranslation process. It is also highly desirable that the design of the common language permit the equipment that the design of the common language permit the equipment system to be designed so that, if desired, information originating in Baudot code form may pass completely through the system to a terminal using Baudot code without any necessity for translation.

---

12. Because of the widespread availability of personnel trained in the use of the standard four-row typewriter keyboard compared to the availability of persons trained on other types of keyboards (e.g. the three-row military teletypewriter keyboard) the choice of code combinations for particular symbols should be made to simplify mechanical or electrical coding when the four-row keyboard is used.

A PROPOSAL FOR CHARACTER CODE COMPATIBILITY

R. W. Bemer, IBM Corporation, White Plains, New York

The emergence of a single standard from a welter of conflicting precedents depends upon two solutions:

1. Selection or development of an adequate and logical standard
2. Phasing out (or peaceful coexistence with) the old varieties

This paper deals with the latter problem and proposes the mechanics for a solution in the area of character codes, as represented by bit combinations.

It appears impossible to reconcile the many different codes in use on paper or magnetic tape such that a particular existing code could be the national or international standard. Because of the wide usage of these various codes, they must be considered parallel standards subject to atrophy through adoption of a single superior code. A simple device that I call the "escape" character will allow as many compatible and graded standards as there are bit combinations in any number of tracks, although it is certainly not desirable to have more of these than absolutely necessary.

Given T character tracks (not feed, parity, or control tracks), there are 2^T possible code combinations. Normally, these are all assigned to specific characters or controls. I propose that one of these combinations, the same one for all standards, be reserved as an "escape" character. This is to be excluded from every such set of characters assigned.

Regarding the choice of this character, it is unwise to use a null, or absence of punches or bits. Furthermore, it is quite possible that the physical permutation of tracks on tape will not be in direct correspondence to the bit pattern of internal storage in a computer or data processing device. The only code that avoids this difficulty is the completely punched combination, or all ones in the bit structure.

Let us make provision for this "escape" combination to interrupt normal decoding of a stream of characters. It will say, in effect, that "The next T-bit combination is to be considered a numeric identifier of a particular standard." From then on, until interrupted by a character in that set, all combinational T-bit characters will be interpreted according to that standard. Shifting from one standard to another is therefore dynamic. A great additional advantage of such a scheme is that many messages in several different codes may be adjoined on the stream of transmission. In hardware, the "escape" character can be made to interrupt to set relays or other switching devices to select one of a variety of readers or decoders.

Another way to view this mechanism is as a two-character shift symbol, comprised of a common symbol and an identifier number. If such numbers are limited to exclude the "escape" combination itself, then two or more "escape" characters in sequence will permit a still further variety of standards (or technically, sub-standards). This is a recursive property.

Admittedly there is some difficulty in applying this principle to most existing codes. For example, the Baudot 5-track code for paper tape already has the "escape" combination assigned to Letter Shift. Fieldata code has this same drawback, at least in the August 1959 version, in that this code is assigned to the letter Z and to Backspace. Although the 5-track teletype codes are virtually impossible to reconcile in this manner, I suggest that all future codes of 6 or more tracks carry this "escape" provision as a built-in safeguard against obsolescence.

NMAH 310/1

(EIA Feb 1960)

January 1960

6984:16-25

	NCR 304	704-709	BENDIX 1-0 TYPER	ALWAC III P-TAPE	HONEYWELL 1000	" 800	BENDIX G-15 (CA-2)	PERSEUS	705	7070 TAPE	1401	650 TAPE	305 RAMAC	DATATRON 220 P-TAPE	1620 P-TAPE	ALWAC III FLEXOWRITER	HIDAC 101 P-TAPE	USS-80	UNIVAC	1103 TYPECODE	BENDIX G-15	LGP-30 FLEXOWRITER	LINCOLN WRITER - UC	LINCOLN WRITER - LC	RCA 501 - BIZMAC	GAMMA 60	FERRANTI PROPOSAL	FIELDATA-MOBIDIC	STRETCH - IBM (SUBSET)
00	0	→	→	b̸	0	0	b̸	0	SM		b̸		b̸	→	→	→	0	0	i		Ⓢ	RD	[illegible]	0	b̸			≈	b̸
1	1	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	Δ	T	-	Z	Σ	1	_	A		ɿ	→
2	2	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	–	Ⓢ	CR	0	|	2	‡	B	CR	%	∧
3	3	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	o	Ⓢ	TAB	SP	‖	3	(	C	LF	◊	↑
4	4	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	1	Ⓢ	Ⓢ	LC	/	4	)	D	TAB	◭	∨
5	5	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	#	2	H	Ⓢ	B	X	5	"	E	BSP	Δ	↓
6	6	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	+	3	N	•	L 1	#	6	:	F	SFT	A	>
7	7	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	)	4	M	Ⓢ	‾ _	→	7	$	G	RO	B	<
10	8	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	5	5		w	U n	<	8	%	H		C	[
1	9	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	6	6	L	x	Y	>	9	;	I		D	]
2	@				'	'		10	0	→	→	→				A	+	7	7	R		* 2	—	_	&	J		E	←
3	¢	=	=	#	#	,		11	#	→	→	→	→	→	=	B	–	8	8	G		= +	□	○	'	K		F	¬
4	b̸	'		@	→	→		12	@	→	→	→	→	→	→	C	•	9	9	I		CS			–	L		G	%
5	&				+	&		Ⓢ		Ⓢ	Ⓢ			Ⓢ		D	,	(	'	P		R			*	M		H	◊
6	•					•		Ⓢ		Ⓢ	Ⓢ					E	(	;	&	C		" 3			•	N		I	#
7	,					+		Ⓢ	TM	→	→					F	)	'	(	V					CS	⊙		J	@
20	–	+	+	&	b̸	b̸		–	b̸	b̸	Ⓢ	b̸	0	→	→	G	S	&	r	E	O	CR	m	A	PC	P	o	K	+ &
1	A	→	→	→	→	→	→	→	/	→	→	→	→	→	→	H	A	A	,	Z	1	I	C	B	LS	Q	|	L	$
2	B	→	→	→	→	→	→	→	S	→	→	→	→	→	→	I	B	B	•	D	2	Δ 4	V	C	/	R	2	M	*
3	C	→	→	→	→	→	→	→	T	→	→	→	→	→	→		C	C	;	B	3		q	D	o	S	3	N	/
4	D	→	→	→	→	→	→	→	U	→	→	→	→	→	→		D	D	A	S	4	BS	r	E	1	T	4	⊙	,
5	E	→	→	→	→	→	→	→	V	→	→	→	→	→	→		E	•	B	Y	5	D	t	F	2	U	5	P	'
6	F	→	→	→	→	→	→	→	W	→	→	→	→	→	→		F	:	C	F	6	% 5	w	G	3	V	6	Q	A
7	G	→	→	→	→	→	→	→	X	→	→	→	→	→	→		G		D	X	7		x	H	4	W	7	R	B
30	H	→	→	→	→	→	→	→	Y	→	→	→	→	→	→		H	E	E	A	8	TAB	i	I	5	X	8	S	C
1	I	→	→	→	→	→	→	→	Z	→	→	→	→	→	→		I	F	F	W	9	N	y	J	6	Y	9	T	D
2	⌑	⊕	>	•	CR	CR		Ⓢ	‡	→	→	→			‡		J	G	G	J	u	$ 6	z	K	7	Z		U	E
3	Δ	•	•		•	#		↓	,	→	→	→	→	→	→		K	H	H	9	v		?	L	8	•		V	F
4		)	)		•	□		↓	%	→	→	→	→	→	(		L	I	I	U			∪	M	9	,		W	G
5			;		(	(		↓		Ⓢ	Ⓢ						M		#	Q		M	∩	N	,	'		X	H
6			∧		)	)		↓		Ⓢ	Ⓢ			Ⓢ		‾ _	N		¢	K	y	π 7	f	⊙	#	;		Y	I
7				□				Ⓢ	CM	TSH	TSM					J	⊙		⊙	⊙	j		λ	P	CN	|		Z	J

Ⓢ = UNASSIGN SPECIAL CHARACTER

6-BIT CODES vs. SYMBOLS

1 of 7

	NCR 304	704-709	BENDIX I-O TYPER	ALWAC III P-TAPE	HONEYWELL 1000	" 800	BENDIX G-15 (CA-2)	PERSEUS	705	7070 TAPE	1401	650 TAPE	305 RAMAC	DATATRON 220 P-TAPE	1620 P-TAPE	ALWAC III FLEXOWRITER	HIDAC 101 P-TAPE	USS-80	UNIVAC	1103 TYPECODE	BENDIX G-15	LGP-30 FLEXOWRITER	LINCOLNWRITER-UC	LINCOLNWRITER-LC	RCA 501-BIZMAC	GAMMA 60	FERRANTI PROPOSAL	FIELDATA-MOBIDIC	RETCH 184 (SUBSET)
40	+	-	→	→	→	→		.	-	→	→	→	→	→	→	K	X	-	t			'	α	Q	A	0		)	K
1	J	→	→	→	→	→	→	→	→	→	→	→	→	→	→	L	P	J	"			ρ	Δ	R	B	1	A	-	L
2	K	→	→	→	→	→	→	→	→	→	→	→	→	→	→	M	Q	K	l	' .		Σ 8	P	S	C	2	B	+	M
3	L	→	→	→	→	→	→	→	→	→	→	→	→	→	→	N	R	L	)	ⓢ			∈	T	D	3	C	<	N
4	M	→	→	→	→	→	→	→	→	→	→	→	→	→	→	⊙	S	M	J	: =			ℓ	U	E	4	D	=	⊙
5	N	→	→	→	→	→	→	→	→	→	→	→	→	→	→	P	Y	$	K	ⓢ		E	⊃	V	F	5	E	>	P
6	⊙	→	→	→	→	→	→	→	→	→	→	→	→	→	→	Q	CR	*	L	' ,		(9	β	W	G	6	F	ω	Q
7	P	→	→	→	→	→	→	→	→	→	→	→	→	→	→	R	Z		M	ⓢ			Λ	X	H	7	G	$	R
50	Q	→	→	→	→	→	→	→	→	→	→	→	→	→	→	lc	LS	N	N	‾ |			λ	Y	I	8	H	*	S
1	R	→	→	→	→	→	→	→	→	→	→	→	→	→	→	uc	U	⊙	⊙	ⓢ		U	~	Z	J	9	I	(	T
2	%	θ	CR		-	=		ⓢ	θ	→	→					CS	V	P	P	1		F	{	(	K	+	J	"	U
3	£	$	→	→	→	→		↓	$	→	→	→	→	→	→	CP	W	Q	Q				}	)	L	-	K	:	V
4	$	*	→	→	→	→		↓	*	→	→	→	→	→	→	TAB		R	R	/ +			≡	+	M	(	L	?	W
5	(		TAB		¢	¢		↓		ⓢ	ⓢ			ⓢ		CR			$			T	=	-	N	)	M	!	X
6	)							↓		ⓢ	ⓢ			ⓢ		BS			*	-		G	'	,	⊙	X	N	,	Y
7	/							ⓢ		Δ	Δ								?	ⓢ			*	.	P	/	⊙	⊕	Z
60	*	ƀ	°		&			ƀ	&	→	→	→	→	→	+	°)		ƀ	Σ	8	⊙				Q	=	P	0	0
1	#	/	→	→	→	→		S	A	→	→	→	→	→	→	/ ,		/	β	ⓢ		H			R	≠	Q	1	1
2	S	→	→	→	→	→	→	T	B	→	→	→	→	→	→	S		S	:	5		J			S	>	R	2	2
3	T	→	→	→	→	→	→	U	C	→	→	→	→	→	→	T		T	+						T	<	S	3	3
4	U	→	→	→	→	→	→	V	D	→	→	→	→	→	→	U		U	/	4					U	Δ	T	4	4
5	V	→	→	→	→	→	→	W	E	→	→	→	→	→	→	V		,	S			C			V	&	U	5	5
6	W	→	→	→	→	→	→	X	F	→	→	→	→	→	→	W		%	T	6		K			W	π	V	6	6
7	X	→	→	→	→	→	→	Y	G	→	→	→	→	→	→	X			U						X	Σ	W	7	7
70	Y	→	→	→	→	→	→	Z	H	→	→	→	→	→	→	Y		V	V	3					Y	%	X	8	8
1	Z	→	→	→	→	→	→	ⓢ	I	→	→	→	→	→	→	Z		W	W			A			Z	£	Y	9	9
2	ℓ	ǂ			:	:		↓	⊕	→	→			ⓢ		* Δ		X	X	7		Q			EF	$	Z	'	.
3	a	,	→	→	→	;		↓	.	→	→	→	→	→	→	' *		Y	Y						ED	*		;	-
4		(	(	%	→	→		↓	⌑	→	→	→	→	→	)	STOP		Z	Z	2					●			/	
5			SP		½	½		↓		ⓢ	ⓢ								%			S			>			.	
6								↓		ⓢ	ⓢ					' ;			=			W			<			□	
7			DEL					ⓢ	ǂ	→	→			ⓢ		. :			δ	ⓢ		DEL			DEL		DEL	BSP	

6-BIT CODES VS. SYMBOLS

2 OF 2

R.W. BEMER   REVISED 8 JAN '60   6984:17a-25

6. Output media
7. Running time
8. Decimal places required for accuracy

e. Errors. Error stops or other stops are listed with reference to the appropriate section of the flow chart. The listing also tells what should be done in case of an error stop.

IV. Flow Chart—Problem-oriented rather than computer-oriented and using a standard set of symbols.

V. The Program Itself. This section consists of the following items:

a. Language and program steps: a listing of the complete program for at least one computer

b. Cross references from one program step to other steps not in sequence

c. Operating instructions including plugboard wiring diagrams where necessary

VI. Sample Problems. Included in the sample problems are the required input data and the desired output format showing the results of the problem.

VII. Notation. This is a list of the nomenclature used in the text and of the symbols in the source program listing.

VIII. Literature References.

IX. Illustrations. Drawings and other illustrative material, prepared in a manner to insure good reproduction, is included in this section.

Editor's Comment: While the ACM has never adopted publication standards, it now appears that *program publication standards* for the ACM may be desirable in the near future. In my opinion, not only is the interest in program interchange growing constantly, but computer-independent language development is bringing us closer to the day when widespread publication of truly catholic programs will be an important function of computer-oriented societies.

Mr. Kent's abridgment of the AIChE committee's report avoids machine-oriented details such as punched card formats; it outlines a procedure which is essentially machine-independent, yet can be used for machine-language publication.

The full report describes means by which program publication is announced and plans for distribution and for covering the cost of small-scale publication by the Society. H. S. B.

A Proposal for Character Code Compatibility

R. W. Bemer, *I.B.M. Corporation, White Plains, N. Y.*

The emergence of a single standard from a welter of conflicting precedents depends upon two solutions:

1. selection or development of an adequate and logical standard,
2. phasing out (or peaceful coexistence with) the old varieties.

This paper deals with the latter problem and proposes the mechanics for a solution in the area of character codes, as represented by bit combinations.

It appears impossible to reconcile the many different codes in use on paper or magnetic tape such that a particular code could be the national or international standard. Because of the wide usage of these various codes they must be considered parallel standards subject to atrophy through adoption of a single superior code. A simple device that I call the "escape" character will allow as many compatible and graded standards as there are bit combinations in any number of tracks, although it is certainly not desirable to have more of these than absolutely necessary.

Given T character tracks (not feed, parity, or control tracks), there are 2^T possible code combinations. Normally these are all assigned to specific characters or controls. I propose that *one* of these combinations, the *same* one for all standards, be reserved as an "escape" character. This is to be excluded from every such set of characters assigned.

Regarding the choice of this character, it is unwise to use a *null*, or absence of punches or bits. Furthermore, it is quite possible that the physical permutation of tracks on tape will not be in direct correspondence with the bit pattern of internal storage in a computer or data-processing device. The only code that avoids these difficulties is the completely punched combination, or all *ones* in the bit structure.

Let us make provision for this "escape" combination to interrupt normal decoding of a stream of characters. It will say, in effect, that "The next T-bit combination is to be considered a numeric identifier of a particular standard." From then on, until interrupted by an "escape" character in *that* set, all combinational T-bit characters will be interpreted according to that standard. Shifting from one standard to another is therefore *dynamic*. A great additional advantage of such a scheme is that many messages in several different codes may be adjoined in the stream of transmission. In hardware, the "escape" character can be made to interrupt to set relays or other switching devices to select one of a variety of readers or decoders.

CACM Feb 1960 vol 3 #2

Another way to view this mechanism is as a two-character shift symbol, comprised of a common symbol and an identifier number. If such numbers are limited to exclude the "escape" combination itself, then two or more "escape" characters in sequence will permit a still further variety of standards (or, technically, sub-standards). This is a recursive property.

Admittedly there is some difficulty in applying this principle to most existing codes. For example, the Baudot 5-track code for paper tape already has the "escape" combination assigned to "Letter Shift." Fieldata code has this same drawback, at least in the August 1959 version, in that this code is assigned to the letter *Z* and to Backspace. Although the 5-track teletype codes are virtually impossible to reconcile in this manner, it is suggested that all future codes of 6 or more tracks carry this "escape" provision as a built-in safeguard against obsolescence.

A Terminology Proposal

FRED GRUENBERGER, *The RAND Corporation, Santa Monica, California*

It has been suggested that we need some new words in our industry. For example, consider the following two definitions of the term *random access*:

(1) Access to storage under conditions in which the next position from which information is to be obtained is *in no way* dependent on the previous one.

(2) Feature of certain internal memory systems, *particularly magnetic drum type.*

(In both cases the italics are the author's.)

Definition (1) is taken from the last official ACM glossary (*Communications, ACM*, October 1958). It has been reproduced verbatim in numerous other glossaries. It is probably the most widely accepted definition of the term today.

Definition (2) is from a booklet titled "What Every Businessman Should Know About Electronic Brains," published by a prominent manufacturer of computers. The subtitle of this booklet is "Or Facts That Will Make You a Cocktail Party Expert on Office Automation." This is a surprisingly accurate title.

This term *random access* is one which is getting kicked around a bit too loosely. One sees phrases like "our new disk file is a random access device with an *average access time* of . . .". If definition (1) is correct, such phrases are internally inconsistent.

It is a fair guess that nearly everyone would agree that tape storage is not random access. But we lack a suitable antonym. Further, those who are pushing disk files are eager for a trichotomy, on the grounds that disk storage is more random than tapes, albeit less random than cores (as used in the 704). The locution "more random" is a logical monstrosity, akin to "more pregnant" or "more unique."

We cannot hope for watertight divisions here. For example, it has been pointed out that, in strict adherence to definition (1), the core storage in STRETCH is nonrandom. Yet it is a long way from the time-dependent access one finds in a disk file.

Therefore, the use of the following three terms is suggested:

(1) Constant-time access
(2) Variable-time access
(3) Serial access

Constant-time access would apply to storage in vacuum tubes, cathode ray tubes, magnetic cores, rods, BIAX elements, and the like. Variable-time access would apply to drums, disk files, tape bins, carrousel storages, and so forth. Serial access, of course, applies to tapes.

"Constant" is preferred to "fixed" to include stores like that on the 604, where the time of access is a function of the oscillator frequency at the moment. Constant-time access replaces exactly the term random access as defined in (1) above.

The three terms used in this proposal were devised by Irwin Greenwald.

EDITOR'S COMMENT: If we specify that "access" is to a single word location, and if we ignore the queuing-time variability of partitioned, buffered stores, Fred's examples imply that "Constant-Time Access" word selection involves only the direct application of digital address information (e.g., switching), *at microsecond speeds.* (We must stretch a concept and call cathode-ray beam positioning "switching".) Unfortunately, some slow stores (e.g., one-revolution-per-word multiple-interlace drums) involve essentially *constant* access time, while our term should imply short access time. Perhaps "Prompt Access" would be more lucid.

His "Variable-Time Access" involves at-least-partially-analog address information (e.g., rotating device angular position, moving head location, acoustic pulse position in a stream, etc.

"Serial Access" would seem to be a special case of "Variable-Time Access", for which the selection in sequence of any two different word locations requires the passing over of *all* intervening address locations. H. S. B.

MINUTES OF MEETING
of the TR 24.4 Subcommittee
On Language and Media
Electronics Industry Association

New York, N. Y.
E.I.A. Headquarters

Mar 24 and 25, 1960

Members Present

Henry Tholstrup	Commercial Controls Corp.
George M. Ives	Ramo-Wooldridge
Gordon Burns	Teletype Corp.
Joseph Lisinski	IBM, San Jose, Calif.
John K. Nelson	Western Union
Harry Heyman	Federal Aviation Agency
C. Toeneboehn	Emerson Electric
J. H. Sweer	RCA
H. H. Bosman	IBM, Endicott
L. L. Griffin	Navy Management Office
Richard M. Gottlieb	U. S. Army Signal R&D Labs.
David Haratz	" " "
G. E. Lyons	IBM, Poughkeepsie
Gloria Bullock	Burroughs
R. A. Von Behren	3 M Co., St. Paul, Minn.
R. H. Hunter	National Cash Register
John Pfaff	Burroughs, Paoli
R. A. Skor	IBM, Poughkeepsie
Thomas J. McNamara	Minn. Honeywell
G. E. Lyons	IBM, Poughkeepsie
J. J. Corless	Stelma, Inc.

Haratz

RH Hunter

Shor

1. Meeting was called to order by Mr. Carl Toeneboehn at 10:00 A.M., March 24, 1960.

2. The following criteria were established for a standard language. Where no vote is indicated, the criteria was adopted unanimously.

   A. Alphabetic characters and numeric characters within themselves should be contiguous in monotonically binary increasing order.

Desired	for	12	against	0
Required	for	5	against	6

#AMA
1/a-3

   B. It is more important in a future standard code to put the alphabet contiguously in binary order than it is to achieve maximum compatibility with Hollerith code.

for 8 against 2

   C. All basic code symbols should contain the same number of binary digits.

   D. The basic code should not contain redundancy.

   E. The basic code should be modularly expandible and within each modular level each character shall have the same number of binary digits.

   F. The code should be so designated that the simplest possible tests shall be adequate to distinguish letters from numbers or symbols.

   G. Within the standard code each character will "stand by itself".

   H. The numeric and alphabetics should be contiguous.

for	5	against	1 (Burroughs)
don't care	2	undecided	3

   I. The language should contain an escape character and in any level of the code it should be the same unique binary configuration.

   J. The numerics should follow the alphabetics.

for	4	against	3
don't care	2	undecided	1

   K. A decimal or alphabetic character including zero should not have all its bit positions zero. This does not include the four bit subsett of any language.

   L. Decimal digits should be represented by their binary value in basic four bit subset.

for 9 against 1

3. Any questions regarding language addressed to E.I.A. or this committee should be answered in the following manner:

   A. This committee is working on a standard language which should be available in the near future.

   B. Although the most commonly used language in data processing at this time is known as the IBM 046 or Flexowriter code and the most commonly used communication code is the five track Baudot code, this does not imply that these are ideal codes or the codes that this committee will adopt.

Vote on the use of the preceding statement for use for answering inquiries

other than publication:	for	12	against	1
For use in publications:	for	8	against	4

4. After considerable discussion the following motion by Mr. Gordon Burns, seconded by Mr. Gottlieb, was adopted by a vote of 7 to 6:

   That this committee take the following position:

   A. The action of March 24 transmitting the standard of hole numbering 1 through 8 to the General Standards Committee is not supported by TR 24.4.4 and should be rescinded.

   B. If a proposal is forwarded it should be the proposal that was submitted and favorably voted in the most recent ballot.

   C. Committee TR-24 should hear a first-hand presentation of the pros and cons before taking a position.

5. The next meeting will be held in San Francisco, May 3 and 4.

6. The meeting was adjourned at 1:00 P.M., March 25, 1960.

Respectfully submitted,

Harry Hayman

Harry Hayman

APPENDIX I

The following table shows the only two languages method A and method B which comply with the criteria recommended by TR24.4 during their meeting of March 24 and 25. For comparative purposes the proposed British Standard and Fieldata codes are shown in adjoining columns. Method A is also compatible with the IBM Stretch language.

Binary	Octal	Method A	Method B	Fieldata	British Proposal
000000	00				
000001	01				
000010	02				
000011	03				
000100	04				
000101	05				
000110	06		A	A	
000111	07		B	B	
001000	10		C	C	
001001	11		D	D	
001010	12		E	E	
001011	13		F	F	
001100	14		G	G	
001101	15		H	H	
001110	16		I	I	
001111	17		J	J	
010000	20		K	K	0
010001	21		L	L	1
010010	22		M	M	2
010011	23		N	N	3
010100	24		O	O	4
010101	25		P	P	5
010110	26	A	Q	Q	6
010111	27	B	R	R	7
011000	30	C	S	S	8
011001	31	D	T	T	9
011010	32	E	U	U	
011011	33	F	V	V	
011100	34	G	W	W	
011101	35	H	X	X	
011110	36	I	Y	Y	
011111	37	J	Z	Z	

Binary	Octal	Method A	Method B	Fieldata	British Proposal
100000	40	K	0		
100001	41	L	1	[illegible]	[illegible]
100010	42	M	2	[illegible]	B
100011	43	N	3	[illegible]	C
100100	44	O	4	[illegible]	D
100101	45	P	5	[illegible]	E
100110	46	Q	6	[illegible]	F
100111	47	R	7	[illegible]	G
101000	50	S	8	[illegible]	H
101001	51	T	9	[illegible]	I
101010	52	U		[illegible]	J
101011	53	V		[illegible]	K
101100	54	W		[illegible]	L
101101	55	X		[illegible]	M
101110	56	Y		[illegible]	N
101111	57	Z		[illegible]	O
110000	60	0		0	P
110001	61	1		1	Q
110010	62	2		2	R
110011	63	3		3	S
110100	64	4		4	T
110101	65	5		5	U
110110	66	6		6	V
110111	67	7		7	W
111000	70	8		8	X
111001	71	9		9	Y
111010	72				Z
111011	73				
111100	74				
111101	75				
111110	76	Escape Character	Escape Character		

APR 20 1961

B

ELECTRONIC INDUSTRIES ASSOCIATION

ENGINEERING DEPARTMENT

2260 SALMON TOWER
11 WEST FORTY-SECOND STREET • NEW YORK 36, N. Y.
Telephone: LOngacre 5-0717

TENTATIVE
STANDARDS PROPOSAL

May 25, 1960

SUBJECT: Basic Character Set Code

BACKGROUND: The attached material was formulated through the combined action of the TR-24.4.1 and TR-24.4.3 Sub-committee on Language and Format for Punched Tape and Magnetic Tapes respectively. (continued on next page).

VOTING
PERIOD
EXPIRES: June 25, 1960

NM AH 310/1

F. The code should be so designated that the simplest possible tests shall be adequate to distinguish letters from numbers or symbols.

G. Within the standard code each character will stand by itself. (Its interpretation should not be dependent upon previous characters).

ND H. The numerals should be contiguous with the alphabet.

I. The language should contain an escape character and in any level of the code expansion it should be the same unique binary configuration. (The escape character permits the use of a different language selected by the succeeding character. This is an exception to criteria G.)

(over)

BACKGROUND (Cont'd):

A common input output language is needed to facilitate intercommunication between data processing and related equipments of various makes, therefore, the proposed tentative standard is oriented toward data processing equipment.

During committee discussions considerable emphasis has been placed upon the desireability of a tape language which was logical and not compromised by having been evolved from other languages developed for other storage media or considerations. An examination showed that many of the existing codes were so compromised.

The following criteria was used by the committee to select the code for a basic character set: (These are not necessarily listed in order of importance.)

A. Alphabetic and numeric characters within themselves should be contiguous in monotonically binary increasing order.

B. It is more important in a future standard code to put the alphabet contiguously in binary order than it is to achieve maximum compatibility with the Hollerith card code.

C. All basic code symbols should contain the same number of binary digits.

D. The basic code should not contain error correcting or detection bits or configurations.

E. The basic code should be modularly expandible and within each modular level each character shall have the same number of binary digits.

F. The code should be so designated that the simplest possible tests shall be adequate to distinguish letters from numbers or symbols.

G. Within the standard code each character will stand by itself. (Its interpretation should not be dependent upon previous characters).

ND H. The numerals should be contiguous with the alphabet.

I. The language should contain an escape character and in any level of the code expansion it should be the same unique binary configuration. (The escape character permits the use of a different language selected by the succeeding character. This is an exception to criteria G.)

(over)

J. The numerics should follow the alphabet.

K. A decimal or alphabetic character including zero should not have all its bit positions zero. This does not include the four bit subset of any language.

L. Decimal digits should be represented by the binary value of the basic four bit subset.

Application of these criteria as a test resulted in the selection of the three codes ~~are~~ included in the Appendix as the most compatible. It is obvious that only codes "A" and "B" meet criteria "H". Although code "B" surpasses "A" in its fitness to criteria "F" Code"A" allows room for the addition of symbols etc. Placement of symbols ahead of the alphabetics has historical precedence in the existing large data files. Although code translation is inevitable on the adoption of any standard, this is minor in comparison to the problem of reordering of the data in these files. Code "A" was, therefore, selected for this tentative standard.

APPENDIX

BINARY	CODE A	CODE B	CODE C	BINARY	CODE A	CODE B	CODE C
000000				100000	K	0	
000001				100001	L	1	
000010				100010	M	2	
000011				100011	N	3	
000100				100100	O	4	
000101				100101	P	5	
000110		A	A	100110	Q	6	
000111		B	B	100111	R	7	
001000		C	C	101000	S	8	
001001		D	D	101001	T	9	
001010		E	E	101010	U		
001011		F	F	101011	V		
001100		G	G	101100	W		
001101		H	H	101101	X		
001110		I	I	101110	Y		
001111		J	J	101111	Z		
010000		K	K	110000	0		0
010001		L	L	110001	1		1
010010		M	M	110010	2		2
010011		N	N	110011	3		3
010100		O	O	110100	4		4
010101		P	P	110101	5		5
010110	A	Q	Q	110110	6		6
010111	B	R	R	110111	7		7
011000	C	S	S	111000	8		8
011001	D	T	T	111001	9		9
011010	E	U	U	111010			
011011	F	V	V	111011			
011100	G	W	W	111100			
011101	H	X	X	111101			
011110	I	Y	Y	111110			
011111	J	Z	Z	111111			

Code C = Fieldata
Code "A" = EIA

BASIC CHARACTER SET CODE

1.0 SCOPE

This proposed code standard for a basic character set is to be used as a basis from which common language standards (for punched tape and for magnetic tape) will be developed. These common language standards are intended primarily for use in interconnection between data processing, and computation equipments or other related uses.

1.1 This proposal covers only the alphabet and arabic numerals. It is the intention that additional assignments for punctuation marks, symbols, and control characters will be added to this standard.

1.2 The basic character set is a (6) six binary digit code, which can be expanded modularly to included error detection and control bits. This extension will be the subject of future actions.

2.0 BASIC CHARACTER SET CODE STANDARD

2.1 Each character in the set is represented by a logically ordered configuration of (6) six binary digits.

2.2 The logical order of these binary digits is shown in Table I. The preferred physical and time position on the storage media will be specified later.

2.3 The english alphabet and arabic numeral assignments to the configuration of binary digits is shown in Table I.

TABLE I

BINARY	ASSIGNMENT	BINARY	ASSIGNMENT
000000		100000	K
000001		100001	L
000010		100010	M
000011		100011	N
000100		100100	O
000101		100101	P
000110		100110	Q
000111		100111	R
001000		101000	S
001001		101001	T
001010		101010	U
001011		101011	V
001100		101100	W
001101		101101	X
001110		101110	Y
001111		101111	Z
010000		110000	0
010001		110001	1
010010		110010	2
010011		110011	3
010100		110100	4
010101		110101	5
010110	A	110110	6
010111	B	110111	7
011000	C	111000	8
011001	D	111001	9
011010	E	111010	
011011	F	111011	
011100	G	111100	
011101	H	111101	
011110	I	111110	
011111	J	111111	

June 1, 1960 TR 00.721 (Rev.)

AN EXTENDED CHARACTER SET STANDARD

by

R. W. Bemer and W. Buchholz

(This report is a revision of the report by the same title dated January 26, 1960 and coded TR 00.18000.705.)

IBM

Product Development Laboratory, Data Systems Division
International Business Machines Corporation, Poughkeepsie, New York

NMAH 310 /1

AN EXTENDED CHARACTER SET STANDARD

by R. W. Bemer and W. Buchholz

1. INTRODUCTION

Present IBM keying and printing equipment is designed to handle 48 or so characters and their codes. In order to guide the development of new equipment for keying and printing many more characters, an Extended Character Set (ECS) has been defined and is shown in Fig. 1. This set contains codes for 120 different characters, but there is room for later expansion to up to 256 characters including control characters. In addition, useful subsets have been defined which contain some but not all of these 120 characters and which use the same codes for the selected characters without translation.

It is intended that the design of future equipment conform to this standard code whenever the added versatility of an extended character set is desired. It is not expected that this code will obsolete the investment in equipment and methods on applications where a 48-character set is entirely adequate.

In selecting this character set, a great deal of thought has been given to satisfying a number of requirements. It proved impossible to satisfy all of them with a single ECS, but selecting a single ECS was considered to be an overriding requirement of the future. The need to communicate between data processing installations and the inevitable mixture of applications in a single installation make it more and more desirable to standardize. A standard character set is, of necessity, a compromise set. In any one situation it is always possible to define a better set, and there are obviously some applications demanding a highly specialized character set. Hence we do not consider this set to be ideal, but we do feel that it satisfies a great many of the more common requirements.

The main purpose of this report is to set down the requirements of an ECS as we see them, and to point out how they have or have not been met by this particular set.

2. SIZE OF SET

Present IBM 48-character sets consist of:

(a) 10 decimal digits,
(b) 26 capital letters,
(c) 11 special characters, and
blank.

Because a single set of 11 special characters is not sufficient, there exist several choices of special characters as "standard options".

Since this set is often represented by a 6-bit code, it is natural to try to extend this set to 63 characters and a blank, so as to exploit the full capacity of a 6-bit code. Although the extra 16 characters would be very useful, this step was thought not to reach far enough to justify the development of the new equipment which would be needed.

As a minimum, the new set should also include

(d) 26 lower case letters,

(e) the more important punctuation symbols found on all office typewriters, and

(f) enough mathematical and logical symbols to satisfy the needs of programming languages such as ALGOL.

There is, of course, no definite upper limit on the number of characters. One could go to the Greek alphabet, various type fonts and sizes, etc., and reach numbers well into the thousands. As the set size increases, however, the cost and complexity of equipment goes up and the speed of printing goes down. The actual choice of 120 characters was purely a matter of judgment of what increment over existing sets would be sufficiently large to justify the departure from present codes without including many characters of only marginal value.

3. SUBSETS

Two subsets of 89 and 49 characters are shown in Figs. 2 and 3. The 89-character set (Fig. 2) is aimed at typewriters which, with 44 character keys, a case shift, and a space bar, can readily handle 89 characters. This subset was considered important because input-output typewriters can already print 89 characters without modification, and 44-key keyboards are familiar to many people.

The 49-character subset (Fig. 3) is usable in a printer similar to the IBM 1403.* It represents the conventional set of "commercial" characters in a code which is compatible with the ECS. Thus, in a system equipped for the ECS it would still be possible to do high-volume printing efficiently on jobs (such as bill printing) where the extra characters may not be needed.

---

*Note that the IBM 1403 has available 49 characters including the blank, which is one more than the 48 characters on earlier printing equipment.

Other subsets are easily derived and may prove useful. For example, for purely numeric work, one may wish to construct a 13-character set consisting of the 10 digits and the symbols . and - together with a special blank.

4. EXPANSION OF SET

Future expansion to a set larger than 120 may take place in two ways. One is to assign additional characters to presently unassigned 8-bit codes; allowance should be made for certain control codes which will be needed for communication and other devices and which are intended to occupy the high end of the code sequence. The second method is to define a shift character to "escape" to another character set. Thus, whenever the shift character is encountered, the next character (or group of characters) identifies a new character set, and subsequent codes are interpreted as belonging to that set. Another shift character in that set can be used to shift to a third set, which may again be the first set or a different set. Such additional sets would be defined only if and when there arise applications which require them.

5. CODE

In choosing a code structure, many alternatives were considered. These varied in the size of the "byte" (i.e., the smallest number of information bits grouped together to represent a character) and in the number of bytes which may represent a single (printable) character. Among them were:

single 6-bit byte with shift codes interspersed,

double 6-bit byte = single 12-bit byte (Ref. 1),

single 8-bit byte,

single 12-bit byte for "standard" characters (punched card code) and two 12-bit bytes for other characters.

Some of these codes were attempts to remain, in some measure, compatible with earlier codes so as to take advantage of existing equipment. These attempts were abandoned, in spite of some rather ingenious proposals, because the advantages of partial compatibility were not enough to offset the disadvantages.

The 8-bit byte was chosen for the following reasons:

(a) The full capacity of 256 characters was considered to be sufficient for the great majority of applications for an ECS.

(b) Within the limits of this capacity, a single character is represented by a single byte so that the length of any particular record is not dependent on the coincidence of characters in that record.

(c) 8-bit bytes are reasonably economical of storage space.

(d) For purely numeric work, a decimal digit can be represented by only 4 bits and two such 4-bit bytes can be packed in an 8-bit byte. Although such packing of numeric data is not essential, it is a common practice to increase speed and storage efficiency. (The IBM 7070, for instance, uses an analogous scheme.) Strictly speaking, 4-bit bytes belong to a different code, but the simplicity of the 4- and 8-bit scheme, as compared to a 4- and 6-bit scheme, for example, leads to a simpler machine design and cleaner addressing logic.

(e) Byte sizes of 4 and 8 bits, being powers of 2, permit the computer designer to take advantage of powerful features of binary addressing and indexing to the bit level (Ref. 2, 3).

In this report, the 8 bits of the code are numbered for identification from left to right as 0 (high-order bit) to 7 (low-order bit). "Bit 0" may be abbreviated to B_0, "Bit 1" to B_1, etc.

6. PARITY BIT

For transmitting data, a ninth bit is attached to each byte for parity checking, and it is chosen so as to provide an odd number of one bits. Assuming a one bit to correspond to the presence of a signal, odd parity permits all 256 combinations of 8 bits to be transmitted and to be positively distinguished from the absence of any signal. The parity bit is identified here as B_P.

The parity bit is defined here for purposes of communication between devices and media using the ECS. It is not intended to exclude the possibilities of error correction or other checking techniques within a given device or on a given medium when appropriate.

7. SEQUENCE

High-equal-low comparisons are an important aspect of data processing. Thus, in addition to defining a standard code for each character, one must also define a standard comparison ("collating") sequence. Obviously, the decimal digits must be sequenced from 0 to 9 in ascending order, and the alphabet from A to Z. Rather more arbitrary is the relationship between groups of characters, but the most prevalent convention for the 48 IBM "commercial" characters is, in order:

(Low)		Blank	
	11	Special Characters	. ⌑ & $ * - / , % # @
	26	Alphabet	A to Z
(High)	10	Digits	0 to 9

Fundamentally, the collating sequence of characters should conform to the natural sequence of the binary integers formed by the bits of that code. Thus 0000 0011 should collate below 0000 0100. Few existing codes have this property, and it is necessary then, in effect, to translate to a special internal code during alphanumeric comparisons. This takes extra equipment, extra time, or both. An important objective of the ECS was to obtain a usable collating sequence directly from the code without translation.

A second objective was to preserve the existing convention for the above 48 characters within the new code. This objective has only partly been achieved because of conflicts with other objectives.

The ECS provides the following collating sequence without any translation:

(Low)		Blank	
	43	Special Characters	(see chart)
	52	Alphabet	a A b B c C to z Z
	20	Digits	0 $_0$ 1 $_1$ to 9 $_9$
	4	Special Characters	. : - ?
(High)		All Unassigned Character Codes	

Note that the lower and upper case letters collate in pairs in adjacent positions, following the convention established for directories of names. (There appeared to be no real precedent for the relative position within the pair. Telephone directories generally ignore the case shift for sequencing purposes, even though both upper and lower case letters appear within some names. This convention is not usable in general since each code must be considered unique.)

The difference between the ECS collating sequence and the earlier convention lies only in the special characters. Two of the previously available characters had to be placed at the high end and the remaining special characters do not fall in quite the same sequence with respect to each other. It was felt that the new sequence would be quite usable, and only rarely will it be necessary to re-sort a file in the transition to the ECS code. It is always possible to translate codes to obtain any other sequence, as has to be done with most existing codes.

8. BLANK

The code 0000 0000 is a natural assignment for the blank (i.e., the nonprint symbol which represents an empty character space). Not only should the blank collate below any printable character, but the absence of bits (other than the parity bit) corresponds to the absence of mechanical movement in a print mechanism.

Blank differs, however, from a "null" character, such as the all-ones code found on paper tape. Blank exists as a definite character occupying a definite position on a printed line, in a record, or in a field to be compared. A "null" may be used to delete an erroneous character and it would be completely dropped from a record at the earliest opportunity. Null, therefore, occupies no definite position in a collating sequence. A null has not been defined here, but it could be placed when needed among the control characters.

9. TYPEWRITER KEYBOARD

Because the shift key on existing IBM 24 and 26 keypunches has been used to cause numbers to punch from otherwise alphabetic keys, it is necessary to establish a completely different convention when introducing lower-case letters. It was thought very desirable, therefore, to take advantage of the widespread familiarity with the typewriter keyboard and to capitalize on existing touch-typing skills as much as possible.

The common typewriter keyboard consists of up to 44 keys, and a separate case shift key. To preserve this relationship in the code, the 44 keys are represented by 6 bits of the code (B_1 to B_6) and the case shift by a separate bit (B_7). The case shift was assigned to the lowest-order bit so as to give the desired sequence between lower and upper case letters.

For ease of typing, the most commonly used characters should appear in the lower shift ($B_7 = 0$). This includes the decimal digits and, when both upper and lower case letters are used in ordinary text, the lower-case letters. (This convention is different from that for single-case typewriters presently used in many data processing systems; when no lower-case letters are available, the digits naturally appear in the same shift as the upper-case

letters.) It is recognized that the typewriter keyboard is not the most efficient alphanumeric keyboard possible, but it would be unrealistic to expect a change in the foreseeable future. For purely numeric data, it is always possible to use a 10-key keyboard instead of, or in addition to, the typewriter keyboard.

It was not practical to retain the upper-lower case relationships of punctuation and other special characters commonly found on typewriter keyboards. There is no single convention anyway, and typists are already accustomed to finding differences in this area.

10. DECIMAL DIGITS

The most compact coding for decimal digits is a 4-bit code, and the natural choices for encoding 0 to 9 are the binary integers 0000 to 1001. As mentioned before, two such digits can be packed into an 8-bit byte; for example, the digits 28 in packed form could appear as

$$\underline{0010}\quad\underline{1000}$$

To represent decimal digits unambiguously in conjunction with other ECS characters, they must have a unique 8-bit representation. The obvious choice is to spread pairs of 4-bit bytes into separate 8-bit bytes and insert a 4-bit prefix ("zone"); for example, the digits 28 might be encoded as

$$z_a z_b z_c z_d\,\underline{0010}\qquad z_a z_b z_c z_d\,\underline{1000}$$

where the actual value of the zone bits z_i is immaterial so long as the prefix is the same for all digits.

This requirement conflicted with requirements for collating sequence and for the shift bit. As a result, the 4-bit byte is offset by one bit, and the actual code for 28 is

$$011\underline{0}\ \underline{010}0\qquad 011\underline{1}\ \underline{000}0$$

This compromise retains the binary integer codes 0000 to 1001 in adjacent bit positions, but not in either of the two positions where they appear in the packed format.

The upper-case counterparts of the normal decimal digits are assigned to italicized decimal subscripts.

11. ADJACENCY

The 52 characters of the upper and lower case alphabets occupy 52 consecutive code positions without gaps. For the reasons given above, it was necessary to spread the 10 decimal digits into every other one of 20 adjacent code positions, but the remaining 10 positions are filled with logically related decimal subscripts. The alphabet and digit blocks are also contiguous. Empty positions for additional data and control characters are all consolidated at the high end of the code chart.

This grouping of related characters into solid blocks of codes, without empty slots that sooner or later would be filled with miscellaneous characters, assists greatly in the analysis and classification of data for editing purposes. ⊙rderly expansion is provided for in advance.

12. UNIQUENESS

A basic principle underlying the choice of the ECS is to have only one code for each character and only one character for each code.

Much of the lack of standardization in existing character sets arises from the need for more characters than there are code positions available in the keying and printing equipment. Thus, in the existing 6-bit IBM character codes, the code 001100 may stand for any one of the characters @ or - or ' The ECS was, instead, required to contain all of these characters with a unique code for each.

The opposite problem exists too. Thus - may be represented by either 100000 or 001100 in one of the existing 6-bit codes. Such an embarrassment of riches presents a logical problem when the two codes have in fact the same meaning and can be used interchangeably. No amount of comparing and sorting will bring like items together until one code is replaced by the other everywhere.

In going to a reasonably large ECS, it was necessary to resist a strong temptation to duplicate some characters in different code positions so as to provide equal facilities in nonoverlapping subsets. Instead, every character was chosen to be typographically distinguishable even if the character stands by itself without context. Thus, for programming purposes, it is possible to represent any code, to which a character has been assigned, by its character even when the bit grouping does not have the ordinary meaning of that character (e.g., in operation codes).

In many instances, however, it is possible to find a substitute character which is close enough to a desired character to represent it in a more restricted subset or for other purposes. For example, = (equals) may stand for ← (is replaced by) in an 89-character subset. ⊙r again, if a hyphen is desired that collates below the alphabet, the symbol ~ (a modified tilde) is preferred to the more conventional - (minus).

A long-standing source of confusion has been the distinction between upper-case ⊙h (O) and Zero (0). Some groups have solved this by writing Zero as ∅ . Unfortunately, other groups have chosen to write ⊙h as ϕ. Neither solution is typographically attractive. Instead, it is proposed to modify the upper-case ⊙h by a center dot and to write and print it as ⊙ whenever a distinction is desired.

Serifs are used to distinguish letters (I, l, V, etc.) from other characters (| , 1 , v , etc.). It is suggested the italicized subscripts be underlined when handwritten by themselves, e.g., 5.

13. SIGNS

The principle of uniqueness implies a separate 8-bit byte to represent a + or - sign. Keying and printing equipment also require separate sign characters. This practice is, of course, rather expensive in storage space, but it was considered superior to the ambiguity of present 6-bit codes where otherwise "unused" zone bits in numeric fields are used to encode signs. If the objective is to save space, one may as well abandon the alphanumeric code quite frankly and switch to a 4-bit decimal coding with a 4-bit sign digit or go to the even more compact binary radix.

14. CARD PUNCHING

After considering the possibility of a separate card code for the ECS characters, a code which has the conventional IBM card code as a subset (Ref. 1), it was concluded that it would be better to punch the ECS code directly into the card. This does not preclude also punching the conventional code (limited to 48 characters) in part of the card for use with conventional equipment. In this way, code translation is needed only wherever the conventional card code is used; if a non-ECS code were used, translation would be required for every column if advantage is to be taken of the ECS code in the rest of the system.

The punching convention is as follows:

Card Row	ECS Bit
12	---
11	---
0	---
1	B_P
2	B_0
3	B_1
4	B_2
5	B_3
6	B_4
7	B_5
8	B_6
9	B_7

In addition, both 12- and 11-holes are to be punched in column 1 of every card containing the ECS code, besides a regular ECS character, so as to distinguish an ECS card from cards punched with the conventional code. ECS punching always starts in column 1 and extends as far as desired; a control code "End" (0 1111 1110) has been defined to terminate the ECS code area. Conventional card code punching should be confined to the right end of cards identified with 12-11 punching in column 1.

Since the parity bit is also punched, the ECS area of a card contains a checkable code. Note that "blank" columns in the ECS area still have a hole in the B_P row. If only part of the card is to be punched, however, it is possible to leave the remaining columns on the right unpunched.

15. ACKNOWLEDGMENTS

The Extended Character Set described here was developed jointly by E. G. Law, H. J. Smith, F. A. Williams, and the authors.

16. REFERENCES

1. R. W. Bemer, "A Proposal for a Generalized Card Code for 256 Characters", Communications of the ACM, vol. 2, no. 9; September 1959.

2. F. P. Brooks, Jr., G. A. Blaauw, W. Buchholz, "Processing Data in Bits and Pieces", IRE Transactions on Electronic Computers, EC-8, no. 2; June 1959.

3. W. Buchholz, "Fingers or Fists? (The Choice of Decimal or Binary Representation)", Communications of the ACM, vol. 2, no. 12; December 1959.

BITS	BITS 0-1-2-3							
4-5-6-7	0000	0001	0010	0011	0100	0101	0110	0111
0000	BLANK	[	&	c	k	s	0	8
0001	±	⊃	+	C	K	S	*0*	*8*
0010	→	]	$	d	l	t	1	9
0011	≠	°	=	D	L	T	*1*	*9*
0100	∧	←	*	e	m	u	2	.
0101	{	≡	(	E	M	U	*2*	:
0110	↑	¬	/	f	n	v	3	–
0111	}	√	)	F	N	V	*3*	?
1000	∨	%	,	g	o	w	4	
1001	∀	\	;	G	⊙	W	*4*	
1010	↓	◊	'	h	p	x	5	
1011	‖	|	" ?	H	P	X	*5*	
1100	>	#	a	i	q	y	6	
1101	≧	!	A	I	Q	Y	*6*	
1110	<	@	b	j	r	z	7	
1111	≦	~	B	J	R	Z	*7*	

Fig. 1. 120 Character Set.

BITS	BITS 0-1-2-3							
4-5-6-7	0000	0001	0010	0011	0100	0101	0110	0111
0000	BLANK		&	c	k	s	0	8
0001			+	C	K	S	*0*	*8*
0010			$	d	l	t	1	9
0011			=	D	L	T	*1*	*9*
0100			*	e	m	u	2	.
0101			(	E	M	U	*2*	:
0110			/	f	n	v	3	–
0111			)	F	N	V	*3*	?
1000			,	g	o	w	4	
1001			;	G	⊙	W	*4*	
1010			'	h	p	x	5	
1011			"	H	P	X	*5*	
1100			a	i	q	y	6	
1101			A	I	Q	Y	*6*	
1110			b	j	r	z	7	
1111			B	J	R	Z	*7*	

10d
11d
STERL
IN 4
Bit Co

Fig. 2. 89 Character Set.

BITS 4-5-6-7	BITS 0-1-2-3 0000	0001	0010	0011	0100	0101	0110	0111
0000	BLANK		&				0	8
0001				C	K	S		
0010			$				1	9
0011				D	L	T		
0100			*				2	.
0101				E	M	U		
0110			/				3	–
0111				F	N	V		
1000		%	,				4	
1001				G	O	W		
1010		◊	'				5	
1011				H	P	X		
1100		#					6	
1101			A	I	Q	Y		
1110		@					7	
1111			B	J	R	Z		

Fig. 3. 49 Character Set.

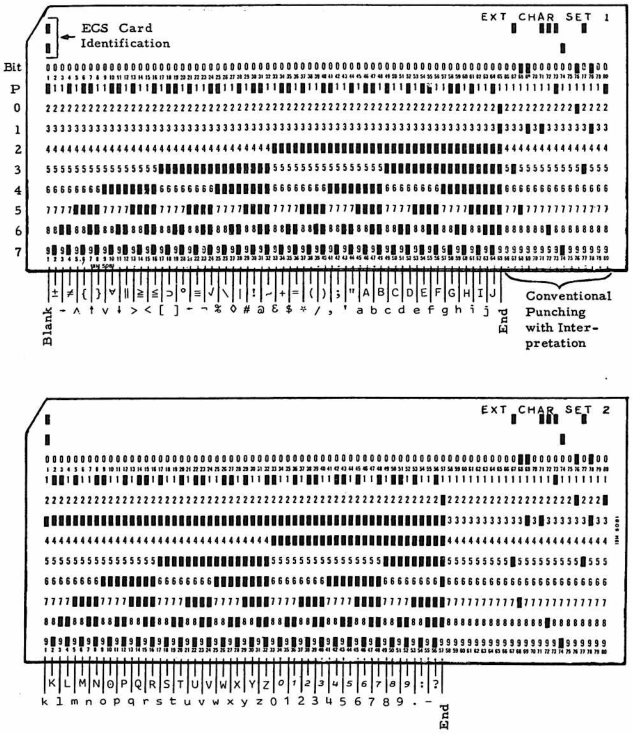

EXT CHAR SET 1
ECS Card Identification
Bit
P
0
1
2
3
4
5
6
7
± ≠ { } ∀ ‖ ≥ ≤ ⊃ ° ≡ √ \ | ! ~ + = ( ) ; " A B C D E F G H I J
→ ∧ ↑ ∨ ↓ > < [ ] ← ¬ % ◊ # @ & $ * / , ' a b c d e f g h i j
Blank
End
Conventional Punching with Interpretation
EXT CHAR SET 2
K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 : ?
k l m n o p q r s t u v w x y z 0 1 2 3 4 5 6 7 8 9 . -
End

Fig. 4. Cards Punched with Extended Character Code.

APPENDIX A

List of ECS Codes and Characters

Code P 0123 4567	Character	Name
1 0000 0000		Blank (Space)
0 0000 0001	±	Plus or minus
0 0000 0010	→	Right arrow (Replaces)
1 0000 0011	≠	Not equal
0 0000 0100	∧	And
1 0000 0101	{	Left brace
1 0000 0110	↑	Up arrow (Start superscript)
0 0000 0111	}	Right brace
0 0000 1000	∨	Or (inclusive)
1 0000 1001	∀	Exclusive Or
1 0000 1010	↓	Down arrow (End superscript)
0 0000 1011	‖	Double lines
1 0000 1100	>	Greater than
0 0000 1101	≧	Greater than or equal
0 0000 1110	<	Less than
1 0000 1111	≦	Less than or equal
0 0001 0000	[	Left bracket
1 0001 0001	⊃	Implies
1 0001 0010	]	Right bracket
0 0001 0011	°	Degree
1 0001 0100	←	Left arrow (Is replaced by)
0 0001 0101	≡	Identical
0 0001 0110	¬	Not
1 0001 0111	√	Square root (Check mark)
1 0001 1000	%	Percent sign
0 0001 1001	\	Left slant (Reverse divide)
0 0001 1010	◊	Lozenge (Diamond) (Note)
1 0001 1011	|	Absolute value (Vertical line)
0 0001 1100	#	Number sign
1 0001 1101	!	Exclamation point (Factorial)
1 0001 1110	@	At sign
0 0001 1111	~	Tilde (Hyphen)
0 0010 0000	&	Ampersand
1 0010 0001	+	Plus sign
1 0010 0010	$	Dollar sign
0 0010 0011	=	Equals
1 0010 0100	*	Asterisk (Multiply)
0 0010 0101	(	Left parenthesis
0 0010 0110	/	Right slant (Divide)
1 0010 0111	)	Right parenthesis
1 0010 1000	,	Comma
0 0010 1001	;	Semicolon
0 0010 1010	'	Apostrophe (Single quote)
1 0010 1011	"	Ditto (Double quote)
0 0010 1100	a	
1 0010 1101	A	
1 0010 1110	b	
0 0010 1111	B	
1 0011 0000	c	
0 0011 0001	C	
0 0011 0010	d	
1 0011 0011	D	
0 0011 0100	e	
1 0011 0101	E	
1 0011 0110	f	
0 0011 0111	F	
0 0011 1000	g	
1 0011 1001	G	
1 0011 1010	h	
0 0011 1011	H	
1 0011 1100	i	
0 0011 1101	I	
0 0011 1110	j	
1 0011 1111	J	
0 0100 0000	k	
1 0100 0001	K	
1 0100 0010	l	
0 0100 0011	L	
1 0100 0100	m	
0 0100 0101	M	
0 0100 0110	n	
1 0100 0111	N	
1 0100 1000	o	
0 0100 1001	O	
0 0100 1010	p	
1 0100 1011	P	
0 0100 1100	q	
1 0100 1101	Q	
1 0100 1110	r	
0 0100 1111	R	
1 0101 0000	s	
0 0101 0001	S	
0 0101 0010	t	
1 0101 0011	T	
0 0101 0100	u	
1 0101 0101	U	
1 0101 0110	v	
0 0101 0111	V	
0 0101 1000	w	
1 0101 1001	W	
1 0101 1010	x	
0 0101 1011	X	
1 0101 1100	y	
0 0101 1101	Y	
0 0101 1110	z	
1 0101 1111	Z	
1 0110 0000	0	Zero
0 0110 0001	$_0$	Subscript zero
0 0110 0010	1	One
1 0110 0011	$_1$	Subscript one
0 0110 0100	2	Two
1 0110 0101	$_2$	Subscript two
1 0110 0110	3	Three
0 0110 0111	$_3$	Subscript three
0 0110 1000	4	Four
1 0110 1001	$_4$	Subscript four
1 0110 1010	5	Five
0 0110 1011	$_5$	Subscript five
1 0110 1100	6	Six
0 0110 1101	$_6$	Subscript six
0 0110 1110	7	Seven
1 0110 1111	$_7$	Subscript seven
0 0111 0000	8	Eight
1 0111 0001	$_8$	Subscript eight
1 0111 0010	9	Nine
0 0111 0011	$_9$	Subscript nine
1 0111 0100	.	Period (point)
0 0111 0101	:	Colon
0 0111 0110	–	Minus sign
1 0111 0111	?	Question mark

NOTE: The character ⌶ has also been used.

APPENDIX B

6-Bit Character Codes in Current IBM Systems

Code	8 Ch. Paper Tape	305, 1620 P. T.	650 Mag. Tape	705; 704 M. T.	7070 Mag. Tape	1401	704 Internal	Code	8 Ch. Paper Tape	305, 1620 P. T.	650 Mag. Tape	705; 704 M. T.	7070 Mag. Tape	1401	704 Internal
00 0000	b	b		Sp		b	0	10 0000	-	-	-	-	-	-	-
00 0001	1	1	1	1	1	1	1	10 0001	J	J	J	J	J	J	J
00 0010	2	2	2	2	2	2	2	10 0010	K	K	K	K	K	K	K
00 0011	3	3	3	3	3	3	3	10 0011	L	L	L	L	L	L	L
00 0100	4	4	4	4	4	4	4	10 0100	M	M	M	M	M	M	M
00 0101	5	5	5	5	5	5	5	10 0101	N	N	N	N	N	N	N
00 0110	6	6	6	6	6	6	6	10 0110	O	O	O	O	O	O	O
00 0111	7	7	7	7	7	7	7	10 0111	P	P	P	P	P	P	P
00 1000	8	8	8	8	8	8	8	10 1000	Q	Q	Q	Q	Q	Q	Q
00 1001	9	9	9	9	9	9	9	10 1001	R	R	R	R	R	R	R
00 1010	PI1		0	0	0	0		10 1010	PI2			$\overset{-}{0}$	$\overset{-}{0}$	$\overset{-}{0}$	$\overset{-}{0}$
00 1011	#	#	#	#	#	#	#	10 1011	$	$	$	$	$	$	$
00 1100	@	@	@	@	@	@	@	10 1100	*	*	*	*	*	*	*
00 1101	PI7							10 1101	PI6						
00 1110	EC1							10 1110	CR						
00 1111	Cor		TM	TM	TM	TM		10 1111	Err				Δ	Δ	
01 0000	0	0	b	b	b	Sp	&	11 0000	&	&	&	&	&	&	b
01 0001	/	/	/	/	/	/	A	11 0001	A	A	A	A	A	A	/
01 0010	S	S	S	S	S	S	B	11 0010	B	B	B	B	B	B	S
01 0011	T	T	T	T	T	T	C	11 0011	C	C	C	C	C	C	T
01 0100	U	U	U	U	U	U	D	11 0100	D	D	D	D	D	D	U
01 0101	V	V	V	V	V	V	E	11 0101	E	E	E	E	E	E	V
01 0110	W	W	W	W	W	W	F	11 0110	F	F	F	F	F	F	W
01 0111	X	X	X	X	X	X	G	11 0111	G	G	G	G	G	G	X
01 1000	Y	Y	Y	Y	Y	Y	H	11 1000	H	H	H	H	H	H	Y
01 1001	Z	Z	Z	Z	Z	Z	I	11 1001	I	I	I	I	I	I	Z
01 1010	PI3		RM	RM	RM	RM	$\overset{+}{0}$	11 1010	SP1			$\overset{+}{0}$	$\overset{+}{0}$	$\overset{+}{0}$	RM
01 1011	,	,	,	,	,	,	.	11 1011	.	.	.	.	.	.	,
01 1100	%	%	%	%	%	%	¤	11 1100	¤	¤	¤	¤	¤	¤	%
01 1101	PI4					WS		11 1101	PI5						
01 1110	Skip							11 1110	SP2						
01 1111	EC2				SM	SM		11 1111	TF			GM	GM	GM	

NOTES:

1. b stands for blank.
2. Sp stands for special control character.
3. The special characters shown in the above chart are those of the "commercial" set (A). For "scientific" computing, character substitutions (sets F and H) are usually made for codes representing certain symbols in the A set:

Commercial	Scientific	
A	F	H
&	+	+
¤	)	)
%	(	(
#	=	=
@	-	'

~~Proposed~~ Addendum to MIL–STD–188A
6-bit alphanumeric code with baudot code equivalents

Data octal equivalent	*Data code*	*Data character*	*Baudot character*	*Baudot code*	
	*54 3210	--------	--------	Case	Code
00	00 0000	Master Space	Blank	E	00000
01	00 0001	Upper Case	Numbers	E	xx0xx
02	00 0010	Lower Case	Letters	E	xxxxx
03	00 0011	Line Feed	Line Feed	E	000x0
04	00 0100	Carriage Ret.	Carriage Ret.	E	0x000
05	00 0101	Space	Space	E	00x00
06	00 0110	A	A	LET	000xx
07	00 0111	B	B	LET	xx00x
10	00 1000	C	C	LET	0xxx0
11	00 1001	D	D	LET	0x00x
12	00 1010	E	E	LET	0000x
13	00 1011	F	F	LET	0xx0x
14	00 1100	G	G	LET	xx0x0
15	00 1101	H	H	LET	x0x00
16	00 1110	I	I	LET	00xx0
17	00 1111	J	J	LET	0x0xx
20	01 0000	K	K	LET	0xxxx
21	01 0001	L	L	LET	x00x0
22	01 0010	M	M	LET	xxx00
23	01 0011	N	N	LET	0xx00
24	01 0100	O	O	LET	xx000
25	01 0101	P	P	LET	x0xx0
26	01 0110	Q	Q	LET	x0xxx
27	01 0111	R	R	LET	0x0x0
30	01 1000	S	S	LET	00x0x
31	01 1001	T	T	LET	x0000
32	01 1010	U	U	LET	00xxx
33	01 1011	V	V	LET	xxxx0
34	01 1100	W	W	LET	x00xx
35	01 1101	X	X	LET	xxx0x
36	01 1110	Y	Y	LET	x0x0x
37	01 1111	Z	Z	LET	x000x
40	10 0000	)	)	NOS	x00x0
41	10 0001	—	—	NOS	000xx
42	10 0010	+	&	NOS	xx0x0
43	10 0011	<	No Equiv.	----	-----
44	10 0100	=	No Equiv.	----	-----
45	10 0101	>	No Equiv.	----	-----
46	10 0110	——	BREAK	----	BREAK
47	10 0111	$	$	NOS	0x00x
50	10 1000	(*)	Bell	NOS	00x0x
51	10 1001	(	(	NOS	0xxxx
52	10 1010	″	″	NOS	x000x
53	10 1011	:	:	NOS	0xxx0
54	10 1100	?	?	NOS	xxx0x
55	10 1101	!	!	NOS	0xx0x
56	10 1110	,	,	NOS	0xx00
57	10 1111	⊕ (Stop)	Stop	NOS	x0x00
60	11 0000	0	0	NOS	x0xx0
61	11 0001	1	1	NOS	x0xxx

FIGURE 19. *Appendix A.*

*** Indicates the order of transmission in serial form, i.e., ascending order.**

NMAH 310/1

6-bit alphanumeric code with baudot code equivalents—Continued

Data octal equivalent	*Data code*	*Data character*	*Baudot character*	*Baudot code*
62	11 0010	2	2	NOS x00xx
63	11 0011	3	3	NOS 0000x
64	11 0100	4	4	NOS 0x0x0
65	11 0101	5	5	NOS x0000
66	11 0110	6	6	NOS x0x0x
67	11 0111	7	7	NOS 00xxx
70	11 1000	8	8	NOS 00xx0
71	11 1001	9	9	NOS xx000
72	11 1010	'	'	NOS 0x0xx
73	11 1011	;	;	NOS xxxx0
74	11 1100	/	/	NOS xx00x
75	11 1101	.	.	NOS xxx00
76	11 1110	□ Special	No Equiv.	---- -----
77	11 1111	Idle	Idle	Line Idle

FIGURE 19—Continued.

Standards

Some Thoughts on Reconciling Various Character Set Proposals

Edward A. Voorhees, *Los Alamos Scientific Laboratory**

Recently there have appeared several proposals for expanded character sets. In general, these differ in the following respects:

a. the number of characters in the expanded set,
b. the characters selected,
c. card and/or tape code representation, and
d. superscripting and subscripting techniques.

Currently, it seems to be popular to talk in terms of establishing a "standard expanded character set." In the opinion of the author, this will prove to be extremely difficult in practice and undesirable in theory. This does not mean, however, that there should be no attempt to achieve "minimal standard character subsets" for communication purposes and "universal languages."

This paper is primarily concerned with a and b above and proposes for each size character set a "standard character subset" and a "locally chosen subset" (denoted by LC). Such a partition of character sets should overcome most of the objections raised when discussions arise concerning which characters should be included in a character set of a given size. It allows "universal languages" to assume the existence of a standard character subset and, if necessary, require that some or all of the LC's be defined for its usage. It also prevents "scientific" symbols being forced upon "business" applications, and vice versa.

49 CHARACTER SET
(1 LC)

bl 1 2 3 4 5 6 7 8 9 0 A B C D E F G H I J K L M N O P Q R S T U
V W X Y Z - + / * = ( ) ' , . \$ L_c
LA :
IBM ǂ OR OTHER

64 CHARACTER SET
(H.S. BRIGHT)

bl 1 2 ← Same as above → T U
V W ← Same as above → \$: ≡ [ ] V Λ > < ; ¬ ↑ ↓ e u d n
UP NULL
END DOWN

NOTE: In some sense, Bright's character set could (or should) be augmented by the 26 lower case alphabetic characters – making a total of 90. Then from a printing standpoint, he could eliminate the u and d – making 88 characters. (SEE BELOW)

89 CHARACTER SET
(14 LCs)

bl 1 2 ← Same as above → T U
V W ← Same as above → \$: a b c d e f g h i j k l m n o
p q r s t u v w x y z L_c L_c L_c L_c L_c L_c L_c L_c L_c L_c L_c L_c L_c L_c
LA '' ? % → [ ] # < > | _ Δ √ π
IBM '' ? ; & 0 1 2 3 4 5 6 7 8 9
HB Λ V ; ≡ [ ] ¬ < > ↑ ↓ e n UNUSED
IAL Λ V ; ≡ [ ] ¬ < > ↑ ↓ ≤ ≥ $_{10}$ ≠ → ×

NOTE: IAL appears to contain 3 extra characters because * ' and \$ are not included.

120 CHARACTER SET
(27 LCs)

bl 1 2 ← Same as above → T U
V W ← Same as above → o
p q ← Same as above → y z '' ? % → ← [ ] { } < > | √ # V Λ ~ ⊂ L_c L_c L_c
L_c L_c – L_c L_c
· LA C ° Σ _ Δ Π × ∫ ∂ ′ α β γ ϵ ϕ ξ η Γ Λ λ μ Θ π ρ σ ψ ζ
IBM ↑ ↓ ≠ ‖ ≥ ≤ ± ° ≡ ¬ \ ◊ ! @ & ; ∀ 1 2 3 4 5 6 7 8 9 0

133 CHARACTER SET
(LOS ALAMOS)

To the above 120-LA set would be added κ τ Φ υ δ χ ω Ω ^ ∞ ∇ ○ □

---

* University of California, Los Alamos, New Mexico. The work was performed under the auspices of the Atomic Energy Commission.

v3 #7. July 1960

It is felt that if agreement can be reached in area b above, then progress can be made in areas c and d. Indeed, since newer computers are being equipped with increasingly powerful editing facilities, it may be that item c is not of major importance. Item d is primarily a point of esthetics as long as one can edit from one notation into another in a straightforward manner.

Standardization with relation to item a is perhaps not meaningful at this time. It may be adequate, however, to base languages on a stated size set (e.g., the 89 set) and, if desired, a reduced and more heavily "conventionalized" language which is based on a smaller set (e.g., the 49 set). The partitioning of sets in this paper (see figure) is a function of set size.

The various "Local Choices" (LC) which are included are designated as follows:

LA (Los Alamos). Consistent with our 133-character set selection; information obtainable from author (to appear soon).

IBM. Their "H" character set as defined in "An Extended Character Set Standard" by R. W. Bemer and W. Buckholz in *IBM Technical Publication*, TR 00.18000.705. January 26, 1960.

IAL. As defined in "Preliminary Report—International Algebraic Language", *Communications ACM*, *1* (December, 1958), 8–22.

HB (H. S. Bright). As described in his letter to the editor, *Communications ACM*, May, 1959.

It should be observed that each standard subset is totally contained in the next larger standard subset except for the 89-IAL set which omits *, ′, and $ and is therefore incompatible to this extent. Mr. Bright's set has also been proposed for IAL usage.

To avoid confusion, "blank" has been counted as a character in the tables.

EDITOR'S NOTE: Current character sets which fit Voorhees' pattern include the 48-character SHARE alphabets and the 64-character TUG alphabet.—H.S.B.

The Multilingual Terminology Project*

By J. E. HOLMSTROM

Mr. Holmstrom, an engineer and technical linguist, is a consultant in the field of multilingual lexicography. He has previously worked for UNESCO.

One of the most important and useful of the tasks which an international organization concerned with some specialized branch of pure or applied science is particularly well able to perform is the improvement of the technical terminology appertaining to that.

Within any given language, improvement of terminology means promoting the idea that the meaning and scope of every technical term ought to be clearly associated in the minds of every writer and every reader with one defined concept and one only. On the international plane it means, further, promoting the ideal that the patterns of technical terminology in all languages ought to correspond exactly with one another, and that it ought to be easy, on referring to a dictionary, to find for every expression in language X an expression in language Y which has precisely the same meaning and implications, no more and no less.

In practice these desiderata are seldom closely approached and never fully attained, because even where a term has a central meaning which is more or less generally agreed, that meaning is apt to be surrounded by a penumbra of vagueness which overlaps the penumbras of other terms. Further, the ambiguity is increased by the existence of homonyms, of synonyms and still more of "quasisynonyms" which mean almost but not quite the same thing. Even within a single language these factors prevent words from being used as the tools of precision they ideally should be. Where translation is involved, the distortion of the message which a technical author wishes to convey to his readers becomes still worse because not even the central meanings of the words in the two languages which the translator equates with one another in his mind, or finds equated to one another in ordinary dictionaries, really coincide.

---

* This paper is reprinted, with permission, from Bulletin No. 8, dated January 1960, of the Provisional International Computation Centre in Rome. The PICC was established by contract between the United Nations Educational, Scientific and Cultural Organization (UNESCO) and the Istituto Nazionale di Alta Matematica, Rome.

Algorithms

REMARK ON ALGORITHM 16
CROUT WITH PIVOTING (G. Forsythe, *Communications* ACM, September, 1960)

George E. Forsythe
Stanford University, Stanford, California

QUERY

Perhaps the most basic procedure for an ALGOL library of matrix programs is an inner product procedure. The procedure Innerproduct given on page 311 of [1] is fairly difficult to comprehend, and probably poses great difficulties for most translating routines. I merely copied its form in writing a modified inner product routine for [2].

My query is: How *should* one write an inner product procedure in ALGOL?

REFERENCES

1. Peter Naur (editor), J. W. Backus, et al., Report on the algorithmic language ALGOL 60, *Comm. Assoc. Comp. Mach.* *3* (1960), 299–314.
2. George E. Forsythe, Crout with pivoting in ALGOL 60, *Comm. Assoc. Comp. Mach.* *3* (1960), 507–508.

ALGORITHM 19
BINOMIAL COEFFICIENTS

Richard R. Kenyon
Computing Laboratory, Purdue University, Lafayette, Indiana

```
comment            This procedure computes binomial coeffi-
                   cients $C_m^n = n!/m!(n - m)!$ by the re-
                   cursion formula $C_{i+1}^n = (n - 1)C_i^n/(i + 1)$
                   starting from $C_0^n = 1$  ;
integer procedure  C(m, n)  ;
integer            m, n  ;
begin              integer i, a, b  ;
                   a := 1  ;
                   if 2 × m > n then b := n — m else
                     b := m  ;
                   for i := 0 step 1 until b do
                   begin a := (n — i) × a ÷ (i + 1) end
                   C := a
end                Binomial Coefficients
```

Contributions to this department must be in the form stated in the Algorithms Department policy statement (*Communications*, February, 1960) except that ALGOL 60 notation should be used (see *Communications*, May, 1960). Contributions should be sent to J. H. Wegstein, Computation Laboratory, National Bureau of Standards, Washington 25, D.C. Algorithms should be in the Publication form of ALGOL 60 and written in a style patterned after the most recent algorithms appearing in this department.

Although each algorithm has been tested by its contributor, no warranty, express or implied, is made by the contributor, the editor, or the Association for Computing Machinery as to the accuracy and functioning of the algorithm and related algorithm material and no responsibility is assumed by the contributor, the editor, or the Association for Computing Machinery in connection therewith.

ALGORITHM 20
REAL EXPONENTIAL INTEGRAL

S. Peavy
National Bureau of Standards, Washington 25, D.C.

```
real procedure     Expint (x)  ;  real x  ;
comment            $-E_i(-x) = \int_x^\infty (e^{-u}/u)\,du$ is computed for
                   x > 0 by approximation formulas. For
                   0 < x < 1 the approximation is from E. E.
                   Allen, Note 169, MTAC 56, pg 240 (1951).
                   The second approximation formula is for
                   1 ≤ x < ∞ and is from C. Hastings, Jr.
                   "Approximations For Digital Computers"
                   (Princeton University Press, Princeton,
                   New Jersey, 1955). The absolute error
                   ε(x) is |ε(x)| < 2 × 10^-7 for 0 < x < 1
                   and |ε(x)| < 2 × 10^-8 for 1 ≤ x < ∞  ;
begin              real y, w, z  ;
                   if x < 1 then
                   z := (((( .00107857 × x — .00976004) × x
                     + .005519968) × x — .24991055) × x
                     + .99999193) × x — .57721566 — ln(x)
                   else begin
                   y := ((( x + 8.5733287401) × x
                     + 18.059016973) × x + 8.6347608925) × x
                     + .2677737343  ;
                   w := ((( x + 9.5733223454) × x
                     + 25.6329561486) × x
                     + 21.0996530827) × x + 3.9584969228  ;
                   z := exp (—x) /x × (y/w) end
                   Expint := z end
```

Standards

Corrigenda*

Edward A. Voorhees, "Some Thoughts on Reconciling Various Character Set Proposals," *Comm. ACM* 3 (July 1960), 408–9:

The character set chart is reproduced (p. 541) with the correction of two characters in the 120-Character Set. The two corrected characters appear in the third line, fourth from the end, correctly: ⊃; and in the fifth line, the third from the beginning, correctly: ⊂.

---

* These errors were introduced in the production of the July issue; apologies are extended to both the author and the Standards Editor.

v3 #10 Oct 1960

ri+hms

ington 25, D.C.

;
) du computed for
ition formulas. For
mation is from E. E.
.C 56, pg 240 (1954).
atio formula is for
om Hastings, Jr.,
Dig l Computers"
Press, Princeton,
The absolute error
-7 fc) < x < 1
for ≤ x < ∞ ;

x - .00976004) × x
: — 24991055) × x
— . 21566 — ln(x)

7401) × x
: + 347608925) × x

23454) × x
x
: x 3.9584969228 ;
(y/\ end

n dards

ghts on Reconciling
om ACM 3 (July

ced (p. 541) with the
120-Character Set
ar i the third line
and n the fifth line,
y: ⊂.

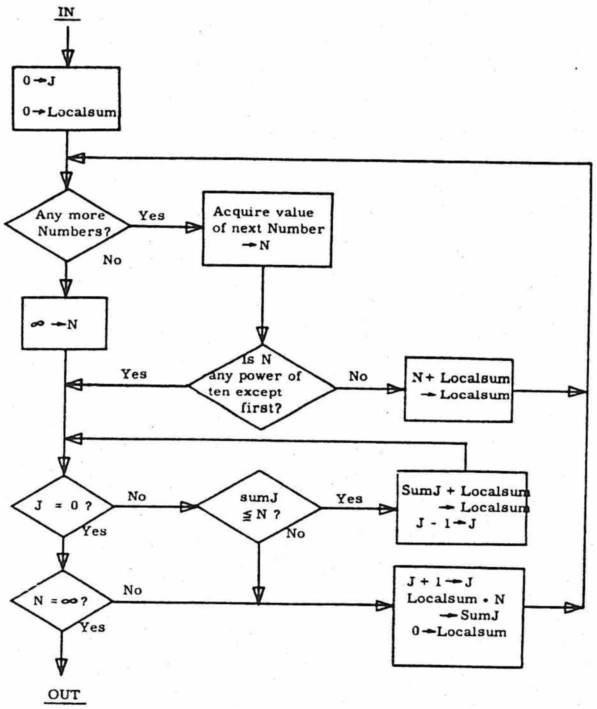
prc ction of the Jo
th and the Standards

(ANOTHER TECHNIQUE . . .)

Evaluating Numbers Expressed as Strings of English Words

Charles J. Swift
Computer Sciences Corp., Inglewood, California

Integer numbers can be expressed in English by a series of words such as "one hundred thirty three million two hundred four." The process indicated by the accompanying flowchart evaluates numbers represented by such a string. To include words up to "billion" requires four terms "Sum$_j$." The multiplications can be done by shifting on a decimal machine.



---

49 CHARACTER SET
(1 LC)

bl 1 2 3 4 5 6 7 8 9 0 A B C D E F G H I J K L M N O P Q R S T U
V W X Y Z – + / * = ( ) ' , . $ └c
LA :
IBM ǂ OR OTHER

64 CHARACTER SET
(H.S. BRIGHT)

bl 1 2 ←——————— Same as above ———————→ T U
V W ←—— Same as above ——→ $: ≡ [ ] ∨ ∧ > < ; ¬ ↑ ↓ e u d n
UP NULL
END DOWN

NOTE: In some sense, Bright's character set could (or should) be augmented by the 26 lower case alphabetic characters – making a total of 90. Then from a printing standpoint, he could eliminate the u and d – making 88 characters. (SEE BELOW)

89 CHARACTER SET
(14 LCs)

bl 1 2 ←——————— Same as above ———————→ T U
V W ←—— Same as above ——→ $: a b c d e f g h i j k l m n o
p q r s t u v w x y z └c └c └c └c └c └c └c └c └c └c └c └c └c └c
LA " ? % → [ ] # < > | _ Δ √ π
IBM " ? ; & 0 1 2 3 4 5 6 7 8 9
HB ∧ ∨ ; ≡ [ ] ¬ < > ↑ ↓ e n UNUSED
IAL ∧ ∨ ; ≡ [ ] ¬ < > ↑ ↓ ≤ ≥ 10 ≠ → ×

NOTE: IAL appears to contain 3 extra characters because * ' and $ are not included.

120 CHARACTER SET
(27 LCs)

bl 1 2 ←——————— Same as above ———————→ T U
V W ←——————— Same as above ———————→ o
p q ← Same as above → y z " ? % → ← [ ] { } < > | √ # ∨ ∧ ~ ⊃ └c └c └c
└c └c — — — — — — — — — — — — — — — — — — └c └c
LA ⊂ ° Σ _ Δ Π × ∫ ∂ ′ α β γ ϵ ϕ ξ η Γ Λ λ μ Θ π ρ σ ψ ζ
IBM ↑ ↓ ≠ ‖ ≥ ≤ ± ° ≡ ¬ \ ◊ ! @ & ; ∀ 1 2 3 4 5 6 7 8 9 0

133 CHARACTER SET
(LOS ALAMOS)

To the above 120-LA set would be added κ τ Φ υ δ χ ω Ω ^ ∞ ∇ ○ □